DB2 Database Development & Management Tool

SQLGate for DB2

User Manual

ⓒ 2011 Antwiz Inc. All rights reserved

Contents of this manual and SQLGate for DB2 program are

protected by copyright law and the protection act of computer program.

■ Technical Support/Customer Support

http://www.antwiz.com

antwiz@antwiz.com

SQLGate for DB2

Table of Contents

1. Introduction of SQLGate for DB2 5

1.1. Introduction 5
1.2. Technical Support 5

2. Product Specification 6

2.1. Major Specification 6
2.2. Operation Environment 7

3. Main Menu 8

3.1. File 8
3.1.1. New 8
3.1.2. New Connection 8
3.1.3. Disconnect 9
3.1.4. Disconnect All 9
3.1.5. Reconnect 9
3.1.6. Open 10
3.1.7. Recent Files 10
3.1.8. Save 10
3.1.9. Save As 10
3.1.10. File Encoding 10
3.1.11. File Type 11
3.1.12. Close Window 11
3.1.13. Close All Windows 11
3.1.14. Print Preview 11
3.1.15. Print 11
3.1.16. Setup Print 11
3.1.17. Exit 11
3.2. Edit 12
3.2.1. Undo 12
3.2.2. Redo 12
3.2.3. Cut 12
3.2.4. Copy 12
3.2.5. Paste 12
3.2.6. Paste from the code 12
3.2.7. Select All 12
3.2.8. Clear All 13
3.2.9. Find and Replace 13
3.2.10. Bookmark 13
3.2.11. Goto line 13
3.2.12. Word Wrap 13
3.2.13. Show Space/Tab/Carriage Return 13
3.2.14. Toggle Folding 13
3.2.15. Comment Block 14
3.2.16. Uncomment Block 14
3.2.17. Convert to 14
3.2.18. Increase Indent 14
3.2.19. Decrease Indent 14
3.2.20. Convert SQL 14
3.2.21. Description Schema 15
3.2.22. Editor Options 16
3.3. View 19
3.3.1. Object Panel 19
3.3.2. SQL History 20
3.3.3. SQL Align Options 23

3.4. Query 24
3.4.1. Create SQL 24
3.4.2. Run SQL 26
3.4.3. Run Current SQL 27
3.4.4. Run SubQuery block 28
3.4.5. Run Current SQL & Modify 29
3.4.6. Execution plan of current SQL 30
3.4.7. Query Description 31
3.4.8. Add to User SQL 31
3.4.9. SQL Align 32
3.5. Create 33
3.5.1. New Table 33
3.5.2. New View 34
3.5.3. New Constraint 35
3.5.4. New Index 36
3.5.5. New Procedures 38
3.5.6. New Function 40
3.5.7. New Trigger 43
3.5.8. New Alias 44
3.5.9. New Sequences 45
3.5.10. New Distinct type 46
3.5.11. New Structured type 46
3.5.12. New Nickname 49
3.6. Tools 50
3.6.1. Reports 50
3.6.2. Export Scripts 52
3.6.3. Export Data 53
3.6.4. Import Data 54
3.6.5. Options 56
3.7. Window 62
3.7.1. Tile Horizontally 62
3.7.2. Tile Vertically 62
3.7.3. Cascade 62
3.7.4. Arrange Icons 62
3.8. Help 63
3.8.1. Help 63
3.8.2. Activation 63
3.8.3. Update Check 63
3.8.4. About SQLGate 64

4. SQL QUERY EDITOR 65

4.1. Multi-Query 65
4.2. Specify background color of text 66
4.3. Run SubQuery block 68
4.4. Multi-Tab 68
4.5. Table Auto Completion 69
4.6. Field Auto Completion 70
4.7. Code Auto Completion 70
4.8. Paste from the code 70
4.9. Execution plan 71

5. Object Explorer 72

5.1. Table 72
5.2. View 74
5.3. Constraint 76
5.4. Index 77
5.5. Procedures 78
5.6. Function 79
5.7. Trigger 80
5.8. Alias 81
5.9. Sequences 82

5.10. Distinct type 83
5.11. Structured type 84
5.12. Nickname 85

6. Query Builder 86

7. ER Design 88

7.1. Table Object 90
7.1.1. Table Object Menu 90
7.2. Relationship 91
7.2.1. Change Relationship object 91
7.3. Group Area 92
7.4. Text Box 92
7.5. Setup Print 92
7.5.1. Page Setup 93
7.6. Find Table Object 93

8. Trivia 94

8.1. Table of shortcut keys 94

1. Introduction of SQLGate for DB2

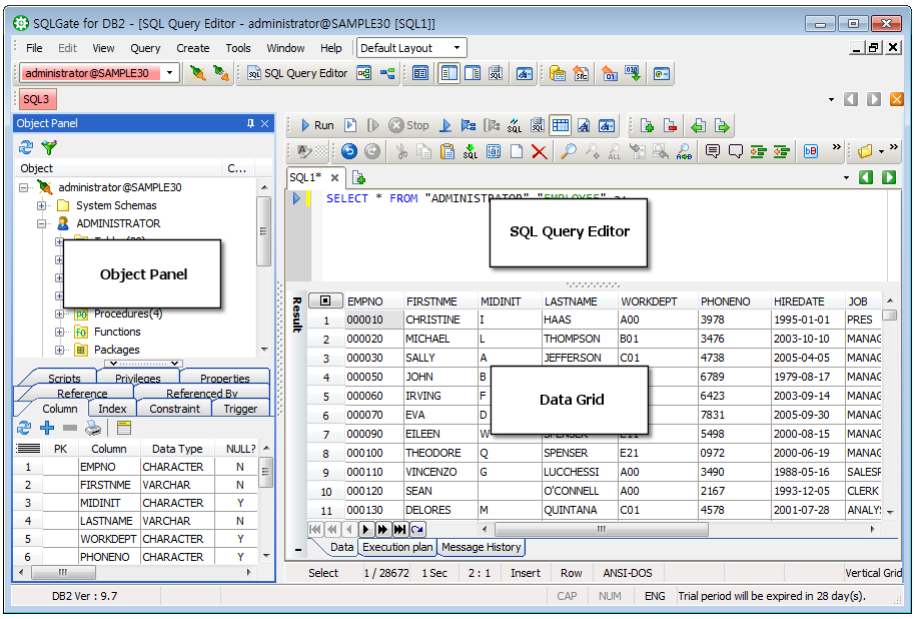
1.1. Introduction

SQLGate for DB2 is a powerful database management tool, which provides you with an easy development and management for DB2 database applications.

It shortens development time and increases business efficiency of database manager and application developer by providing user-friendly interface and various features.

You can create and execute SQL easily because code auto completion, keyword auto completion and table auto completion feature is supported in SQL Query Editor.

Moreover, you can change user interface into multi language version with just simple setting in options because this is created in 7 different languages.



1.2. Technical Support

If you have any questions about this software, refer to this document or post questions on the Technical Forum in company's homepage (http://www.sqlgate.com/en/support/).

Technical Support : +82 2-338-1876

2. Product Specification

2.1. Major Specification

- **Unicode Support**

It supports Unicode. Multilingual can be handled through it.
UTF8, UTF16, AL32UTF8 character sets are supported.

- **User Friendly Interface**

It consists of editor, grid and user friendly interface, which are convenient and powerful features. It satisfies user's needs by providing various styles and options to grid, such as XP, Float, Office and so on.

- **Easy and Fast Query Execution**

Users can create SQL code easily by using SQL Query Editor and check results of queries.
Moreover, users can create faster and more accurate SQL code by using the auto completion feature.

- **Powerful Object Browser**

It helps inquiry, creation, modification and deletion for each user through object browser. Users can inquire easily and quickly by linking with SQL Query Editor.

- **Provision for Tool of Various Features**

Even beginners can operate various database management tasks easily by providing various tools, such as Word Wrap, Bookmark, SQL Conversion , Report, Import/Export Utility and so on.

- **Report Tool to Print as Desired Types**

Users can design directly and print out data by using Report Designer and shorten creation time of various reports.

- **ER Design Feature**

Users can create ERD (entity-relationship diagram) by reversing relationship between tables in ER Design window without installation of separate program.

2.2. Operation Environment

- **Operation Environment**

All of Windows Family OSs

Over Intel Pentium 4

Over 512MB Memory

Over 50MB free hard disk space

- **Configuration**

Features of folders, which are composed program, are as follows.

Bin : Composed of execution files and ini, dat files

HealthCheck : Html files, which are used at HealthCheck

Languages : Folder to support multilingual

Report : Folder to save report samples

Screen : Dll files, which are needed at form configuration

3. Main Menu

3.1. File

3.1.1. New

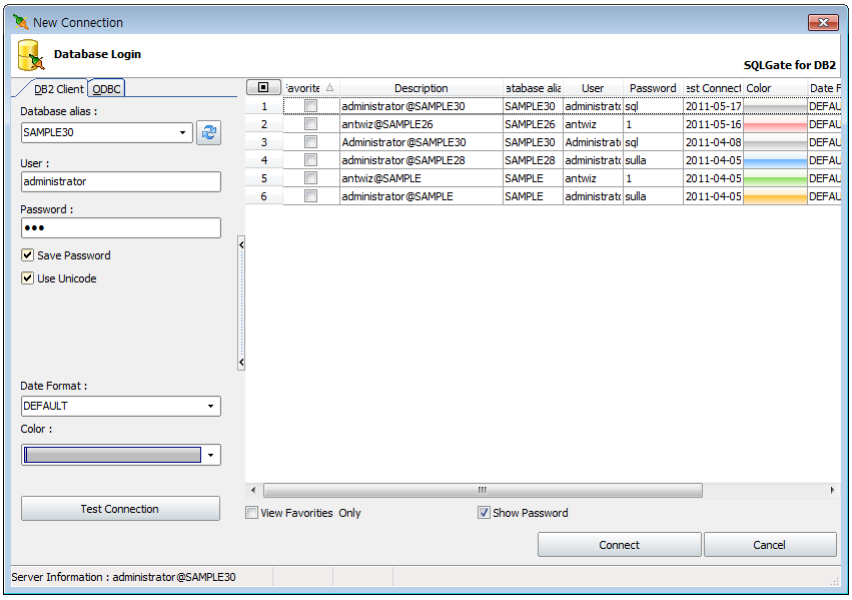
It creates new SQL Query Editor window.

SQL QUERY EDITOR : Select **[File] > [New] > [SQL QUERY EDITOR < Ctrl+N >]** on the main menu or click the **[** **]** button.

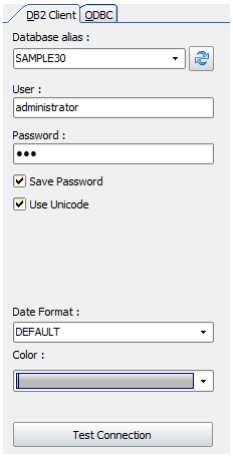
3.1.2. New Connection

Select **[File] > [New Connection...]** on the main menu or click the **[** **]** button.

You can connect through not only DB2 but also ODBC.



- **DB2 Client Tab**



Database alias : Select the alias to connect to.

User: Connection Account

Password : Password

Save Password : Select whether to save password or not.

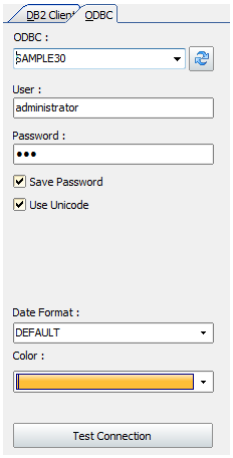
Use Unicode : Use Unicode mode when connect.

Date Format : Date type format.

Color : Select color by session when you connect to several DB2s.

Test Connection : Execute connection test to inputted information.

- **ODBC Tab**



ODBC : Select the alias to connect to.

User: Connection Account

Password : Password

Save Password : Select whether to save password or not.

Use Unicode : Use Unicode mode when connect.

Date Format : Date type format.

Color : Select color by session when you connect to several DB2s.

Test Connection : Execute connection test to inputted information.

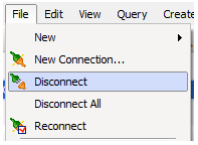
※ If you add a database in "Configuration Support Program" of DB2, the database will be shown in **[New Connection]** window as alias.

※ In addition, when more than one DB2 version of DB2 client are installed, if you try to connect to the database, you must execute "Default DB2 and database client interface select wizard" and select the copy of default DB2.

3.1.3. Disconnect

Disconnects to currently connected DB2 server.

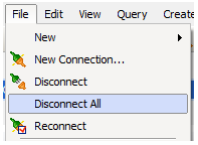
Select **[File] > [Disconnect]** on the main menu.



3.1.4. Disconnect All

Disconnects to currently all connected DB2 servers.

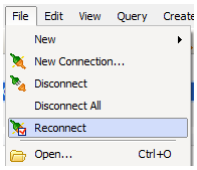
Select **[File] > [Disconnect All]** on the main menu.



3.1.5. Reconnect

Reconnects to currently connected DB2 server.

Select **[File] > [Reconnect]** on the main menu.



3.1.6. Open

If you select **[File] > [Open …< Ctrl+O >]** on the main menu, Open File dialog will be shown.

If you select a file, the file will be opened on SQL Query Editor.

If SQL Query Editor doesn't exist, the file will be opened after executing new SQL Query Editor.

3.1.7. Recent Files

If you select **[File] > [Recent Files]** on the main menu, list of recently opened files is shown.

If you select a file, the file will be opened on SQL Query Editor. If SQL Editor doesn't exist, the file will be opened after executing new SQL Query Editor.

3.1.8. Save

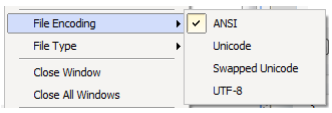
If you select **[File] > [Save < Ctrl+S >]** on the main menu, you can save contents on SQL Query Editor.

3.1.9. Save As

If you select **[File] > [Save As...< Ctrl+Shift+S >]** on the main menu, you can save contents on SQL Query Editor as new name.

3.1.10. File Encoding

If you select **[File] > [File Encoding]** on the main menu, you can convert encoding type of SQL Query Editor into ANSI, Unicode, Swapped Unicode or UTF-8.



3.1.11. File Type

If you select **[File] > [File Type]** on the main menu, you can convert file type of SQL Query Editor into Default, Windows(CR/NL), Macintosh(CR) or Unix(NL).

3.1.12. Close Window

If you select **[File] > [Close Window]** on the main menu, you can close the current window.

3.1.13. Close All Windows

If you select **[File] > [Close All Windows]** on the main menu, you can close all windows.

3.1.14. Print Preview

Select **[File] > [Print Preview]** on the main menu to open preview window.

3.1.15. Print

Select **[File] > [Print...]** on the main menu to print contents on SQL Query Editor.

3.1.16. Setup Print

If you select **[File] > [Printer Setting]** on the main menu, Print Setting/Page Setting will be shown.
You can set printer and page setting

3.1.17. Exit

Select **[File] > [Exit]** on the main menu to close program after disconnecting connected session.

3.2. Edit

3.2.1. Undo

Cancel the just previous task.

Select **[Edit] > [Undo < Ctrl+Z >]** on the main menu.

3.2.2. Redo

Re-execute canceled task.

Select **[Edit] > [Redo < Shift+Ctrl+Z >]** on the main menu.

3.2.3. Cut

Cut off selected contents.

Select **[Edit] > [Cut < Ctrl+X >]** on the main menu.

3.2.4. Copy

Copy selected contents to clipboard.

Select **[Edit] > [Copy < Ctrl+C >]** on the main menu.

3.2.5. Paste

Paste contents in clipboard.

Select **[Edit] > [Paste < Ctrl+V >]** on the main menu.

3.2.6. Paste from the code

Paste contents by extracting SQL statement in contents of clipboard.

Select **[Edit] > [Paste from the code]** on the main menu or press the shortcut key **< Shift+Ctrl+V >**.

Ex) Clipboard : “select * from member”

Paste in the code : select * from member

3.2.7. Select All

Select all texts in editor.

Select **[Edit] > [Select All < Ctrl+A >]** on the main menu.

3.2.8. Clear All

Clear all texts in editor.

Select **[Edit] > [Clear All]** on the main menu.

3.2.9. Find and Replace

Find or Replace contents in editor.

Select **[Edit] > [Find and Replace]** on the main menu.

3.2.10. Bookmark

Add, Move or Delete bookmark in editor.

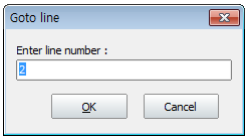
Select **[Edit] > [Bookmark]** on the main menu.

- **[Toggle Bookmark]** : Add bookmark.
- **[Goto Bookmark] :** Move to bookmark.
- **[Clear All Bookmarks]** : Delete all bookmarks.

3.2.11. Goto line

Move to inputted line at the Goto line dialog.

Select **[Edit] > [Goto line... < Ctrl+G >]** on the main menu.



3.2.12. Word Wrap

The text wraps around if it is longer than horizontal area of the editor.

Select **[Edit] > [Word Wrap]** on the main menu.

3.2.13. Show Space/Tab/Carriage Return

Space, Tab and Carriage Return will be shown among texts in editor.

Select **[Edit] > [Show Space/Tab/Carriage Return]** on the main menu.

3.2.14. Toggle Folding

If the folding is set, [+/-] marks will be displayed to collapse or expand editing contents at the left of the editor.

If you click the [-], editing contents will be collapsed and click the [+], they will be expand again.

Select **[Edit] > [Toggle Folding]** on the main menu.

3.2.15. Comment Block

Make selected text in editor into comments. Marked contents as comment will not be executed.

Select **[Edit] > [Comment Block < Ctrl+- >]** on the main menu.

3.2.16. Uncomment Block

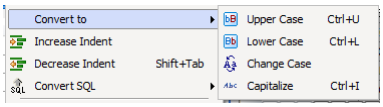
Remove blocked comments of selected texts in editor.

Select **[Edit] > [Uncomment Block < Shift+Ctrl+- >]** on the main menu.

3.2.17. Convert to

Convert selected texts in editor into upper case, lower case, upper/lower case or convert only first character into upper case.

Select **[Edit] > [Convert to]** on the main menu.



3.2.18. Increase Indent

Indent one line in editor. (Shortcut key **< Tab >**)

Select **[Edit] > [Increase Indent]** on the main menu.

3.2.19. Decrease Indent

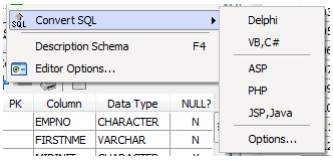
Remove the indentation of one line in editor.

Select **[Edit] > [Decrease Indent < Shift+Tab >]** on the main menu.

3.2.20. Convert SQL

Convert texts in editor into Delphi, Visual Basic, ASP, PHP, JSP code and copy to clipboard.

Select **[Edit] > [Convert SQL]** on the main menu.



3.2.21. Description Schema

Open Description Schema window about the keyword, where a cursor is located in editor.

Select **[Edit] > [Description Schema < F4 >]** on the main menu.

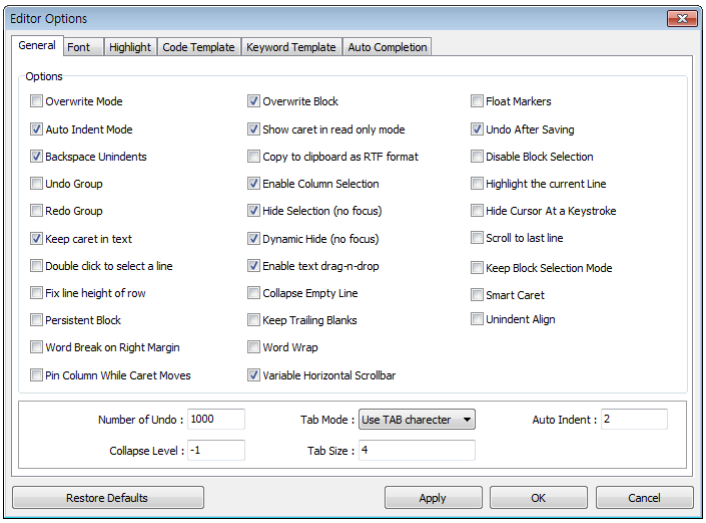
3.2.22. Editor Options

Set options of editor.

Select **[Edit] > [Editor Options]** on the main menu.

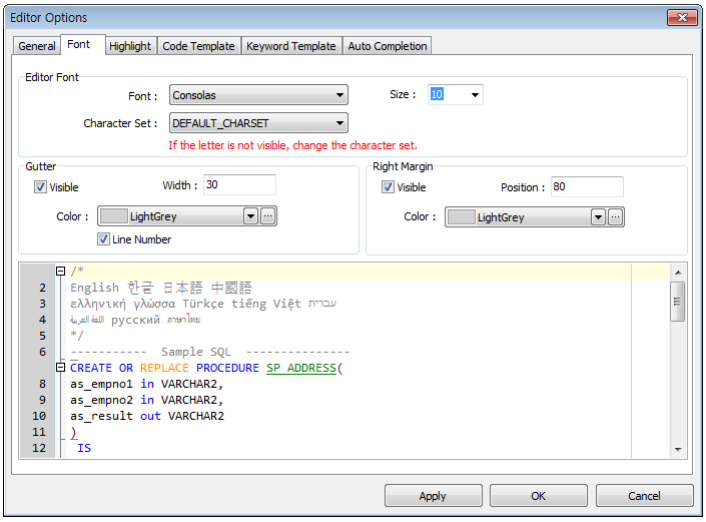
- **General**

Set general options.



- **Font**

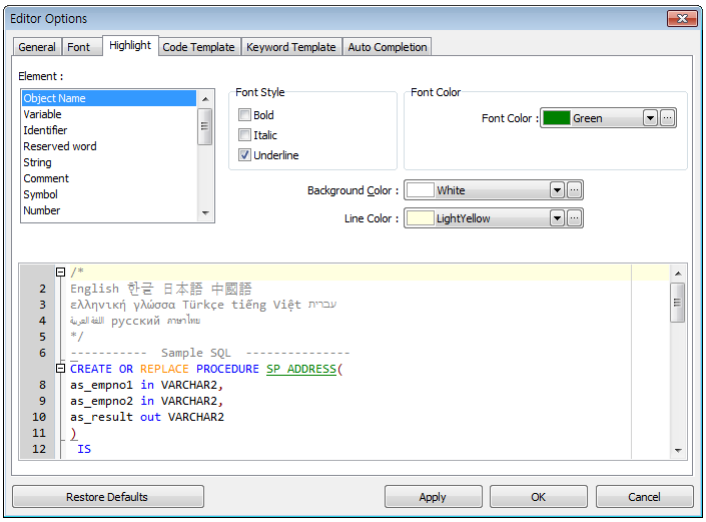
Set font options.



※ If specific text is broken in editor, change the character set.

- **Highlight**

Set highlight options.

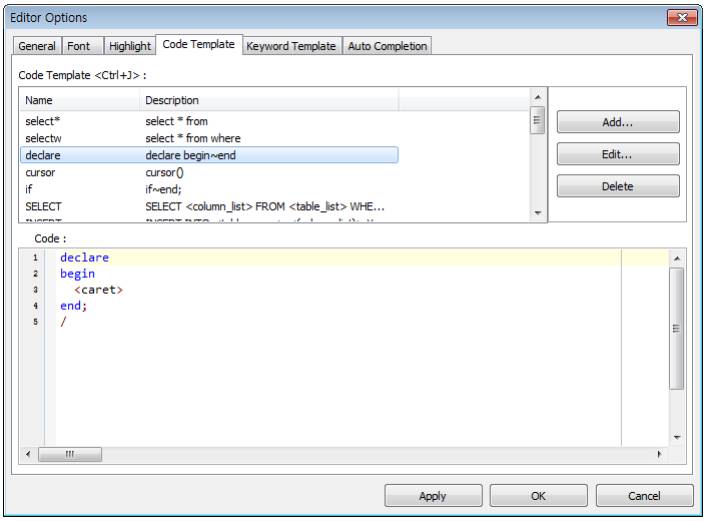


You can change font style, font color and so on after selecting one of elements in **[Element]**.

Moreover, elements will be chosen automatically in **[Element]** when you select some of sentences in the bel ow editor.

- **Code Template**

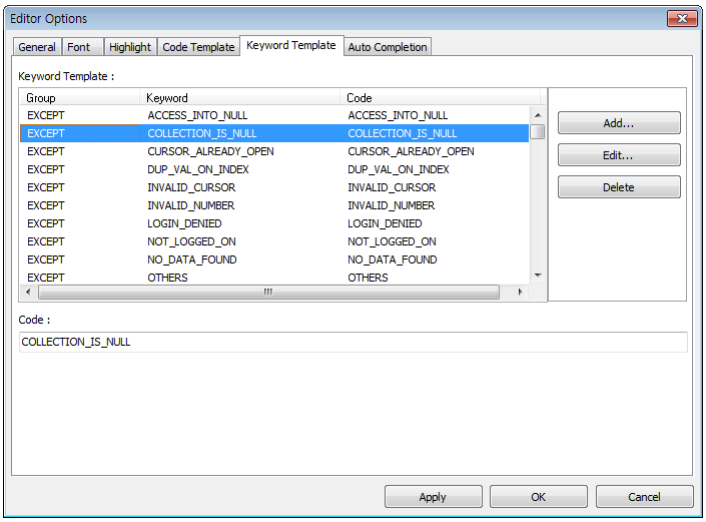
Add, Edit or Delete code template.



※ If you press **< Ctrl+J >** in editor, template will be displayed.

- **Keyword Template**

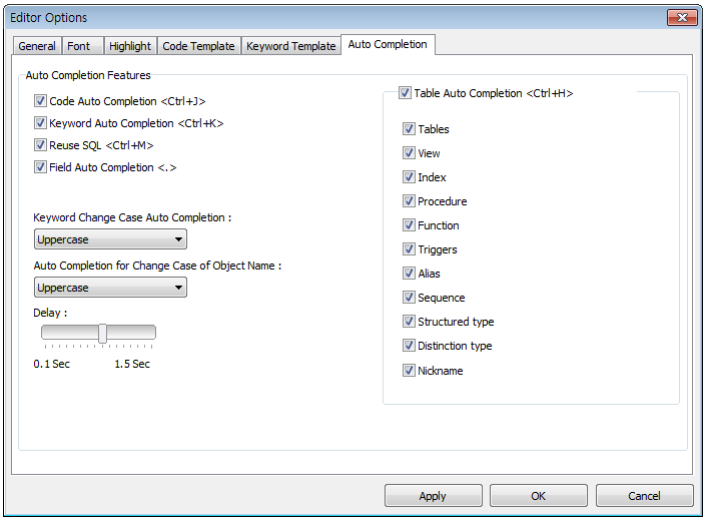
Add, Edit or Delete keyword template.



※ Select **[Table Auto Completion]** in **[Auto Completion]** tab to activate that keyword in SQL Query Editor.

- Auto Completion

Set whether to use Auto Completion feature.



- **Code Auto Completion< Ctrl+J >** : Display contents in code template after auto completion.
- **Keyword Auto Completion< Ctrl+K >** : Keyword Auto Completion will be displayed if you type more than 3 characters.
- **Reuse SQL< Ctrl+M >** : Checked sentences are displayed among contents in SQL history.
- **Field Auto Completion< . >** : Display field list auto completion next to table alias.
- **Keyword Change Case Auto Completion** : Display completed keyword by Keyword Auto Completion in disting uishing upper/lower character.
- **Auto Completion for Change Case of Object Name** : Selected object by Table Auto Completion in distinguis hing upper/lower character.
- **Table Auto Completion < Ctrl+H >** : When a character is inputted after FROM / INTO / SET, the object will be displayed after auto completion.

  ※ You can use it optionally when there are too many objects.

3.3. View

3.3.1. Object Panel

Object panel can be docked at the left or right of the main window and it can search and manage objects.

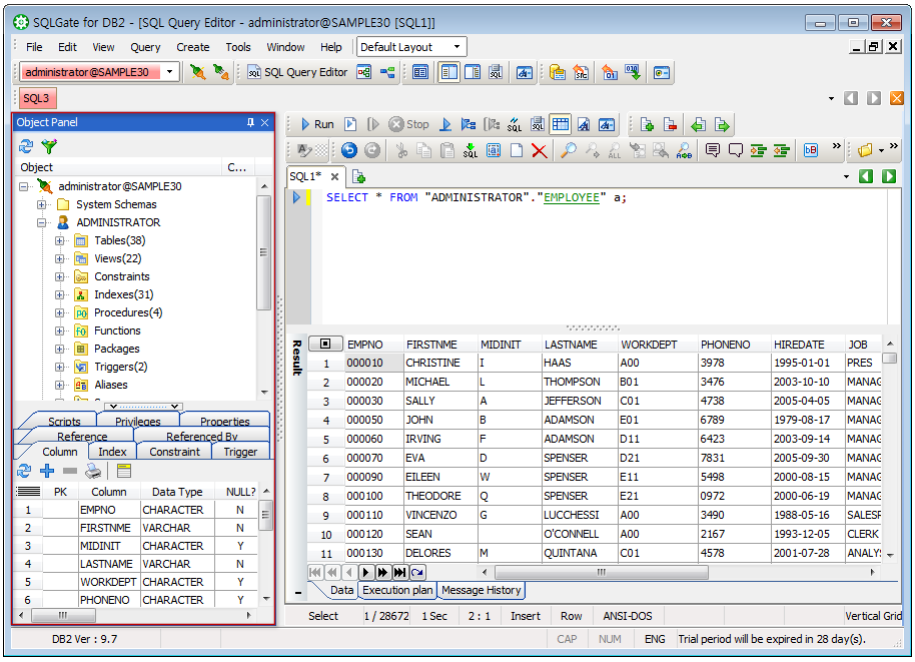
If you select an object, detail information about the object will be shown at the bottom of the window.

If you click the right mouse button after selecting an object, you can add, delete, edit or manage it.

You can add SQL to SQL Query Editor simply by using Object Scripting feature.

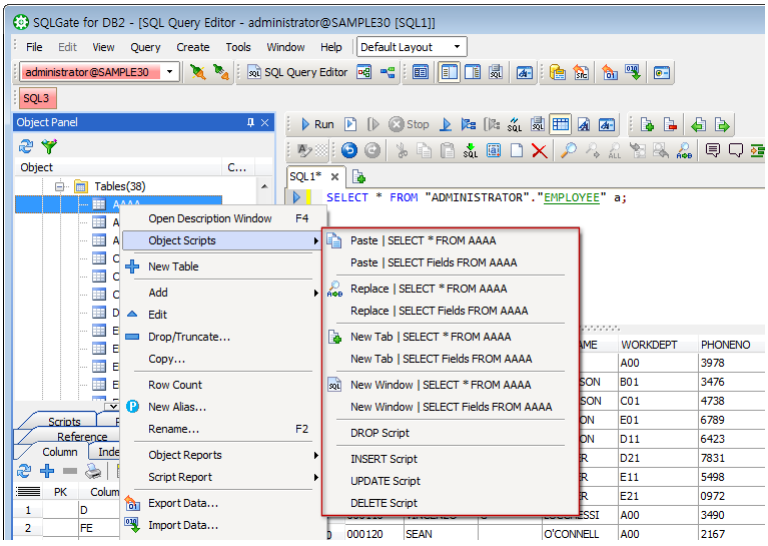
You can create various queries and check results with only simple operation.

Select **[View] > [Object Panel]** on the main menu or click the [ ] button.



- **Object Scripts**

If you click the right mouse button after selecting Table or View objects, you can use **[Object Scripts]** menu, helping SQL creation.



[Paste | Select] : Paste Select script of the selected object to SQL Query Editor.

[Replace | Select] : Change the Select script of the selected object in SQL Query editor.

[New Tab | Select] : Open Select script of the selected object in new tab.

[New Window | Select] : Open Select script of the selected object in new SQL Query Editor window.

[Drop Script]: Paste Drop script of the selected table or view to SQL Query Editor.

[Insert Script] : Paste Insert script of the selected table or view to SQL Query Editor.

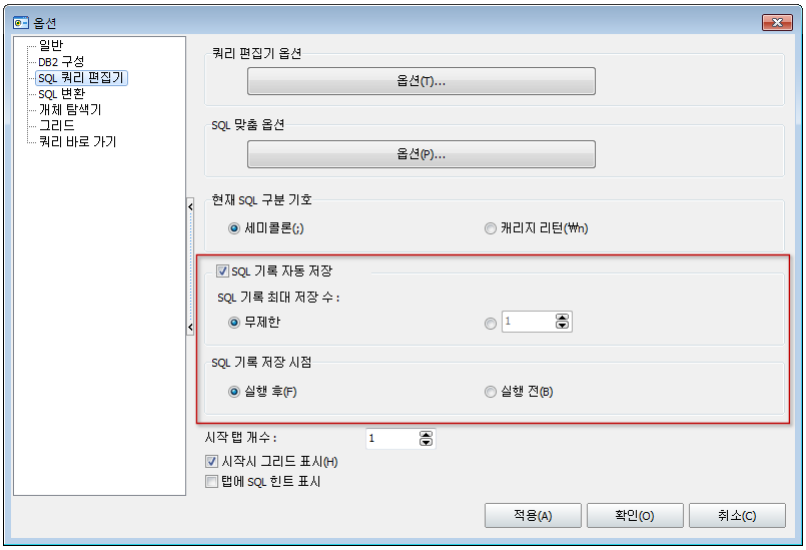
[Update Script] : Paste Update script of the selected table or view to SQL Query Editor.

[Delete Script] : Paste Delete script of the selected table or view to SQL Query Editor.

3.3.2. SQL History

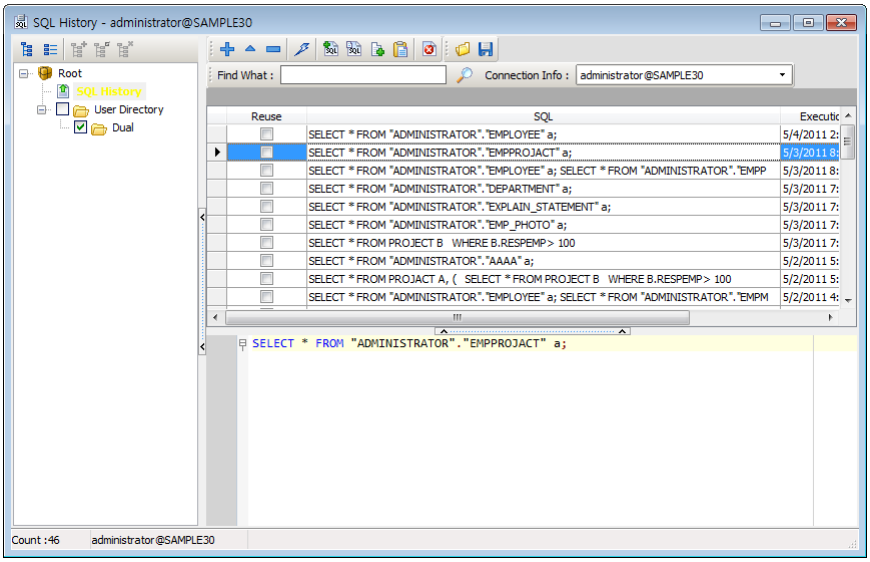
It saves successfully executed SQLs in **[SQL Query Editor]**.

Check **[Tools] > [Options ...] > [SQL QUERY EDITOR]** > **[AUTO SAVE SQL HISTORY]** to save SQL history in Program options.

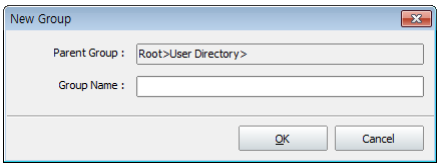


- **AUTO SAVE SQL HISTORY** : When checked, SQL will be saved.
- **MAX SAVE COUNT OF SQL HISTORY** :
  - **Unlimited :** Save all without limitation.
  - **Count :** Save as much as inputted number.
- **STORE POINT OF SQL HISTORY** :
  - **After Execution** : Save only succeeded query commands.
  - **Before Execution** : Save all query commands.

Select **[View] > [SQL History ...< F8 >]** on the main menu or click the **[** **]** button.


SQL History - administrator@SAMPLE30
Root
SQL History
User Directory
Dual
Find What :
Connection Info :
administrator@SAMPLE30
Reuse
SQL
SELECT * FROM "ADMINISTRATOR"."EMPLOYEE" a;
SELECT * FROM "ADMINISTRATOR"."EMPPROJACT" a;
SELECT * FROM "ADMINISTRATOR"."EMPLOYEE" a; SELECT * FROM "ADMINISTRATOR"."EMPP
SELECT * FROM "ADMINISTRATOR"."DEPARTMENT" a;
SELECT * FROM "ADMINISTRATOR"."EXPLAIN_STATEMENT" a;
SELECT * FROM "ADMINISTRATOR"."EMP_PHOTO" a;
SELECT * FROM PROJECT B WHERE B.RESPEMP> 100
SELECT * FROM "ADMINISTRATOR"."AAAA" a;
SELECT * FROM PROJACT A, (SELECT * FROM PROJECT B WHERE B.RESPEMP> 100
SELECT * FROM "ADMINISTRATOR"."EMPLOYEE" a; SELECT * FROM "ADMINISTRATOR"."EMPM
SELECT * FROM "ADMINISTRATOR"."EMPPROJACT" a;
Count :46
administrator@SAMPLE30

You can add folder by selecting **[New Group ...]** on the context menu (by clicking right mouse button) after selecting **User Directory** in the left tree.



DML is saved to **SQL History** group basically.

SQL can move from SQL history list to the other group through drag&drop.

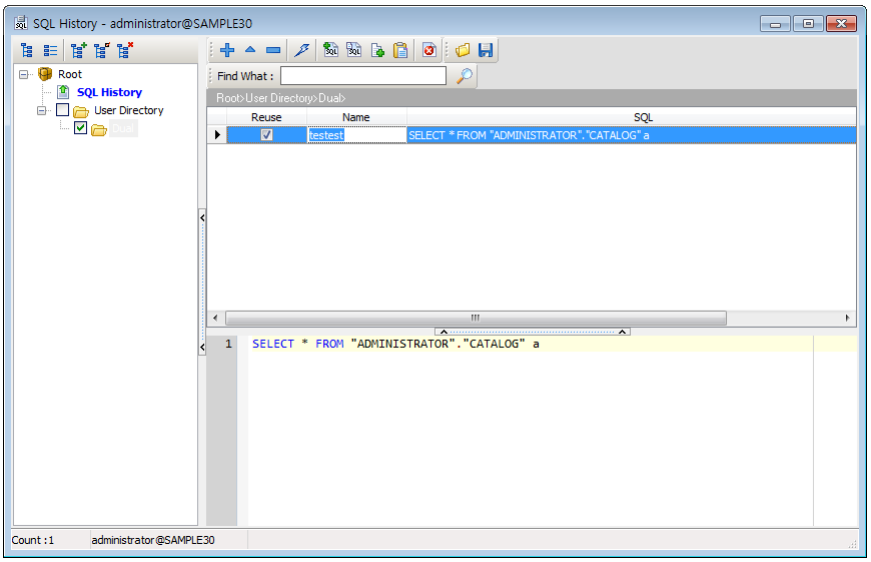
If you double click grid, the selected SQL transfers to SQL Query Editor.

You can save to dat file or open SQL history through the **[Open File]** or the **[Save File]** button.

- **Reuse SQL**

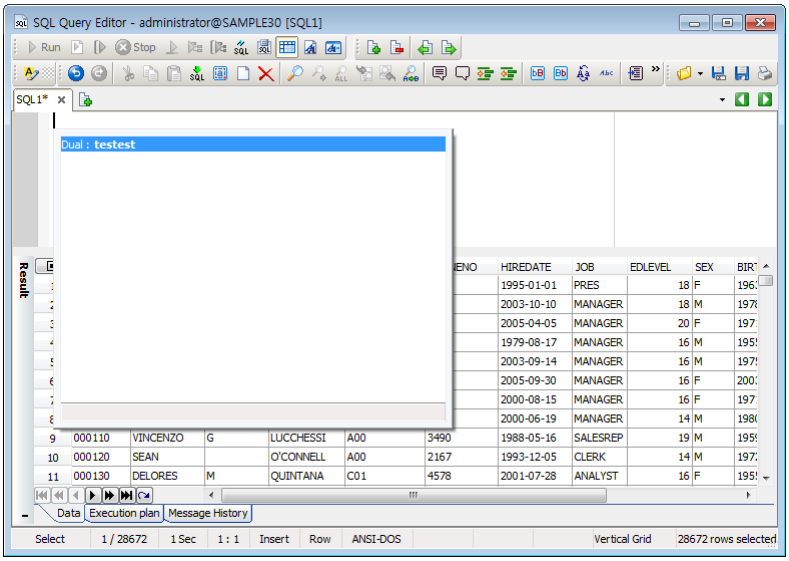
Check the **[Reuse]** checkbox in SQL history, which is included to **[User Directory]** group and its sub groups.

Input wanted characters in **[Name]**.

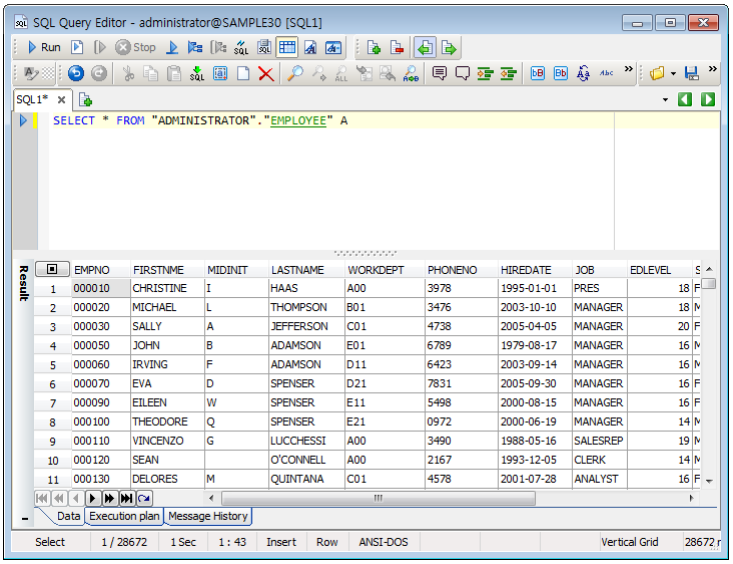


If you click the shortcut key **< Ctrl+M >** in SQL Query Editor, SQL will be called.

Select one in reuse list.



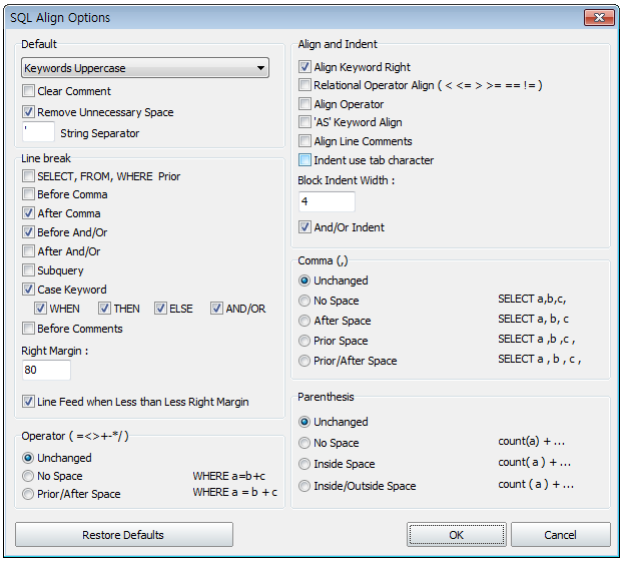
SQL, which is saved in SQL history, will be inputted automatically.


SQL Query Editor - administrator@SAMPLE30 [SQL1]
Run
Stop
SQL1*
SELECT * FROM "ADMINISTRATOR"."EMPLOYEE" A
Result
EMPNO FIRSTNME MIDINIT LASTNAME WORKDEPT PHONENO HIREDATE JOB EDLEVEL
1 000010 CHRISTINE I HAAS A00 3978 1995-01-01 PRES 18
2 000020 MICHAEL L THOMPSON B01 3476 2003-10-10 MANAGER 18
3 000030 SALLY A JEFFERSON C01 4738 2005-04-05 MANAGER 20
4 000050 JOHN B ADAMSON E01 6789 1979-08-17 MANAGER 16
5 000060 IRVING F ADAMSON D11 6423 2003-09-14 MANAGER 16
6 000070 EVA D SPENSER D21 7831 2005-09-30 MANAGER 16
7 000090 EILEEN W SPENSER E11 5498 2000-08-15 MANAGER 16
8 000100 THEODORE Q SPENSER E21 0972 2000-06-19 MANAGER 14
9 000110 VINCENZO G LUCCHESSI A00 3490 1988-05-16 SALESREP 19
10 000120 SEAN O'CONNELL A00 2167 1993-12-05 CLERK 14
11 000130 DELORES M QUINTANA C01 4578 2001-07-28 ANALYST 16
Data
Execution plan
Message History
Select
1 / 28672
1 Sec
1 : 43
Insert
Row
ANSI-DOS
Vertical Grid
28672

3.3.3. SQL Align Options

It is option for alignment or indentation feature of created SQL.

Select **[View] > [SQL Align Options...]** on the main menu.



- **Default** : Set Upper/Lower case, Clear Comment, Remove Blank and Separator of keyword.
- **Line break** : Set location of line to change.
- **Align & Indent** : Set alignment of Keyword, Operator, AS Keyword, Comment, AND/OR and so on.
- **Comma** : Set blank before/after comma.
- **Parenthesis** : Set blank before/after parenthesis.
- **Operator** : Set blank before/after operator.

3.4. Query

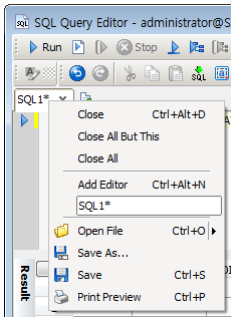
3.4.1. Create SQL

If you select **[File] > [New] > [SQL Query Editor < Ctrl+N >]** on the main menu or click the **[ ]**button, SQL Query Editor window will be shown.

- **SQL QUERY EDITOR Tab**

You can use up to 20, if you use tab. You can use SQL Query Editor and grid in each tab separately.

If you click the right mouse button in tab, context menu will be shown.



[Close < Ctrl+Alt+D >] : Close the tab.

[Close All But This] : Close all tabs except the selected tab.

[Close All] : Close all tabs.

[Add Editor < Ctrl+Alt+N >] : Add a tab.

[Open File < Ctrl+O >] : Open a file in the current tab.

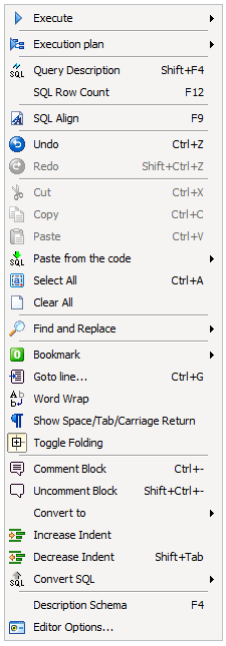
[Save As...] : Save SQL in the current tab as another name.

[Save < Ctrl+S >] : Save SQL in the current tab.

[Print Preview < Ctrl+P >] : Preview contents to print in the current tab.

Context Menu of SQL QUERY EDITOR

If you click the right mouse button in SQL Query Editor, following context menu will be shown.



[Execute] : Execute SQL in editor / Execute SQL of the selected text

[Query Description < Shift+F4 >] : Field definition by analyzing SQL statement

[SQL Row Count < F12 >] : Count records where cursor is located at

[SQL Align < F9 >] : Align all SQLs in editor

[Undo < Ctrl+Z >] : Cancel execution

[Redo < Shift+Ctrl+Z >] : Re-execute

[Cut < Ctrl+X >] : Cut off

[Copy < Ctrl+C >] : Copy

[Paste < Ctrl+ V>] : Paste

[Paste from the code] : Extract SQL from other program codes

[Select All < Ctrl+A >] : Select all

[Clear All] : Clear all

[Find and Replace] : Find, Find Next, Find All, Find from first , Find the last, Replace

[Bookmark] : Toggle bookmark

[Goto line… < Ctrl+G >] : Move directly to the line by inputting line number

[Word Wrap] : Wrap around automatically

[Show Space/Tab/Carriage Return] : Show space, tab and carriage return character

[Toggle Folding] : Turn on/off folding feature

[Comment Block < Ctrl+- >] : Set comment

[Uncomment Block < Shift+Ctrl+- >] : Release comment

[Convert to] : Convert to upper/lower character

[Increase Indent] : Indent the line where cursor is located

[Decrease Indent < Shift+Tab >] : Unindent line where cursor is located

[SQL CONVERT] : Convert into Delphi, Visual Basic, ASP, PHP and JSP code

[Description Schema < F4 >] : Show Description window of keyword where cursor is located

[Editor Options] : Open editor options window

3.4.2. Run SQL

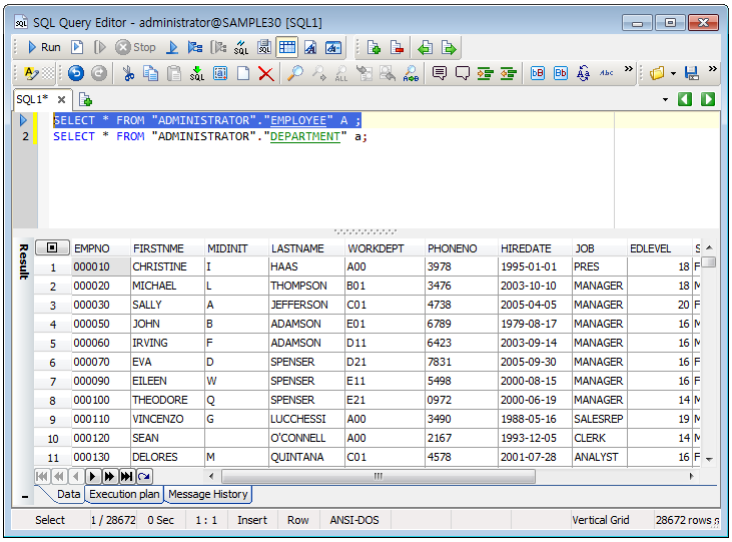
Executes created SQL in SQL Query Editor. If several SQLs are executed, it is separated with semicolon (;).

Select **[Query] > [Run SQL]** on the main menu or press the [ ] button or **< F5 >** key.

In case of Select statement, execution results are displayed as grid type in **[Data]** tab at the bottom

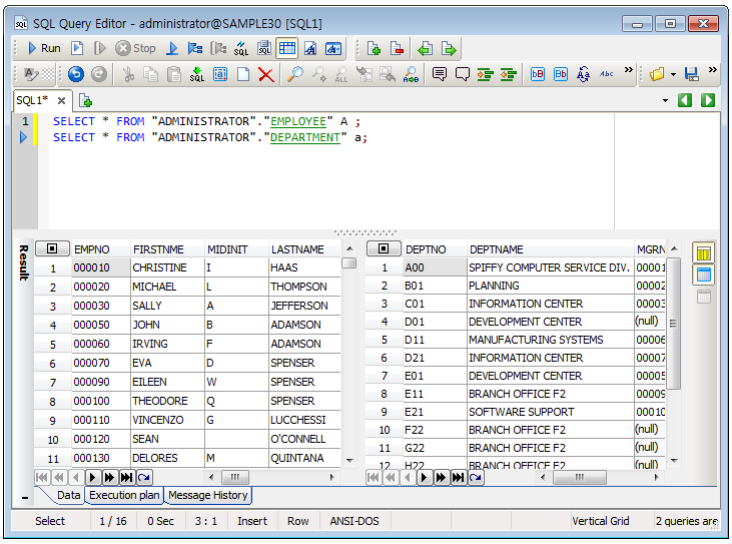
In case of non Select statement, execution results are displayed in **[Message History]** tab.

Click the [ ]button or press **< ESC >** key to cancel execution.


SQL Query Editor - administrator@SAMPLE30 [SQL1]
Run
Stop
SQL1*
SELECT * FROM "ADMINISTRATOR"."EMPLOYEE" A ;
SELECT * FROM "ADMINISTRATOR"."DEPARTMENT" a;
Result
EMPNO FIRSTNME MIDINIT LASTNAME WORKDEPT PHONENO HIREDATE JOB EDLEVEL
000010 CHRISTINE I HAAS A00 3978 1995-01-01 PRES 18
000020 MICHAEL L THOMPSON B01 3476 2003-10-10 MANAGER 18
000030 SALLY A JEFFERSON C01 4738 2005-04-05 MANAGER 20
000050 JOHN B ADAMSON E01 6789 1979-08-17 MANAGER 16
000060 IRVING F ADAMSON D11 6423 2003-09-14 MANAGER 16
000070 EVA D SPENSER D21 7831 2005-09-30 MANAGER 16
000090 EILEEN W SPENSER E11 5498 2000-08-15 MANAGER 16
000100 THEODORE Q SPENSER E21 0972 2000-06-19 MANAGER 14
000110 VINCENZO G LUCCHESSI A00 3490 1988-05-16 SALESREP 19
000120 SEAN O'CONNELL A00 2167 1993-12-05 CLERK 14
000130 DELORES M QUINTANA C01 4578 2001-07-28 ANALYST 16
Data
Execution plan
Message History
Select
1 / 28672
0 Sec
1 : 1
Insert
Row
ANSI-DOS
Vertical Grid
28672 rows

If you select text in SQL Query Editor, only selected text will be executed.

If there is no selected block, all SQLs will be executed.

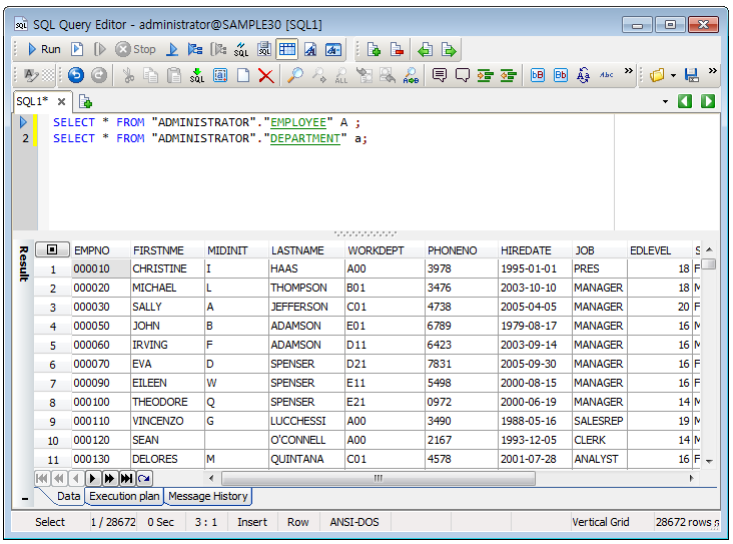


If you click the [ ] button, you can see horizontal grid.

3.4.3. Run Current SQL

Executes SQL, where cursor is located, only in SQL Query Editor.

Select **[Query] > [Run Current SQL < Ctrl+F5 > < Ctrl+Enter >]** on the main menu or press the **[ ]** button.


SQL Query Editor - administrator@SAMPLE30 [SQL1]
Run
Stop
SQL1*
SELECT * FROM "ADMINISTRATOR"."EMPLOYEE" A ;
SELECT * FROM "ADMINISTRATOR"."DEPARTMENT" a;
Result
EMPNO FIRSTNME MIDINIT LASTNAME WORKDEPT PHONENO HIREDATE JOB EDLEVEL
1 000010 CHRISTINE I HAAS A00 3978 1995-01-01 PRES 18
2 000020 MICHAEL L THOMPSON B01 3476 2003-10-10 MANAGER 18
3 000030 SALLY A JEFFERSON C01 4738 2005-04-05 MANAGER 20
4 000050 JOHN B ADAMSON E01 6789 1979-08-17 MANAGER 16
5 000060 IRVING F ADAMSON D11 6423 2003-09-14 MANAGER 16
6 000070 EVA D SPENSER D21 7831 2005-09-30 MANAGER 16
7 000090 EILEEN W SPENSER E11 5498 2000-08-15 MANAGER 16
8 000100 THEODORE Q SPENSER E21 0972 2000-06-19 MANAGER 14
9 000110 VINCENZO G LUCCHESSI A00 3490 1988-05-16 SALESREP 19
10 000120 SEAN O'CONNELL A00 2167 1993-12-05 CLERK 14
11 000130 DELORES M QUINTANA C01 4578 2001-07-28 ANALYST 16
Data
Execution plan
Message History
Select
1 / 28672
0 Sec
3 : 1
Insert
Row
ANSI-DOS
Vertical Grid
28672 rows

※ Run SQL ('SELECT * FROM "ADMINISTRATOR"."EMPLOYEE" a ;') in the line, where cursor is located.

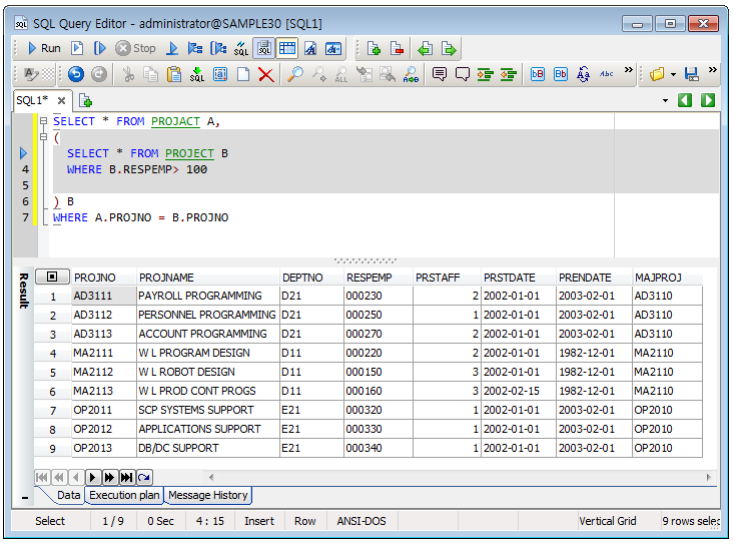
3.4.4. Run SubQuery block

Run subquery in the line, where cursor is located, in SQL Query Editor.

Subquery can be executed even it is not in block. Font color of subquery in editor is changed automatically.

Also the **[Run SubQuery block < Alt+F5 >]** button is activated.

Select **[Query] > [Run SubQuery block]** on the main menu or press the **[ ]** button or **< Alt+F5 >** key.


SQL Query Editor - administrator@SAMPLE30 [SQL1]
Run
Stop
SQL1*
SELECT * FROM PROJACT A,
(
SELECT * FROM PROJECT B
WHERE B.RESPEMP> 100
) B
WHERE A.PROJNO = B.PROJNO
Result
PROJNO PROJNAME DEPTNO RESPEMP PRSTAFF PRSTDATE PRENDATE MAJPROJ
1 AD3111 PAYROLL PROGRAMMING D21 000230 2 2002-01-01 2003-02-01 AD3110
2 AD3112 PERSONNEL PROGRAMMING D21 000250 1 2002-01-01 2003-02-01 AD3110
3 AD3113 ACCOUNT PROGRAMMING D21 000270 2 2002-01-01 2003-02-01 AD3110
4 MA2111 W L PROGRAM DESIGN D11 000220 2 2002-01-01 1982-12-01 MA2110
5 MA2112 W L ROBOT DESIGN D11 000150 3 2002-01-01 1982-12-01 MA2110
6 MA2113 W L PROD CONT PROGS D11 000160 3 2002-02-15 1982-12-01 MA2110
7 OP2011 SCP SYSTEMS SUPPORT E21 000320 1 2002-01-01 2003-02-01 OP2010
8 OP2012 APPLICATIONS SUPPORT E21 000330 1 2002-01-01 2003-02-01 OP2010
9 OP2013 DB/DC SUPPORT E21 000340 1 2002-01-01 2003-02-01 OP2010
Data
Execution plan
Message History
Select
1 / 9
0 Sec
4 : 15
Insert
Row
ANSI-DOS
Vertical Grid
9 rows sele

※ Background color of subquery is changed automatically.

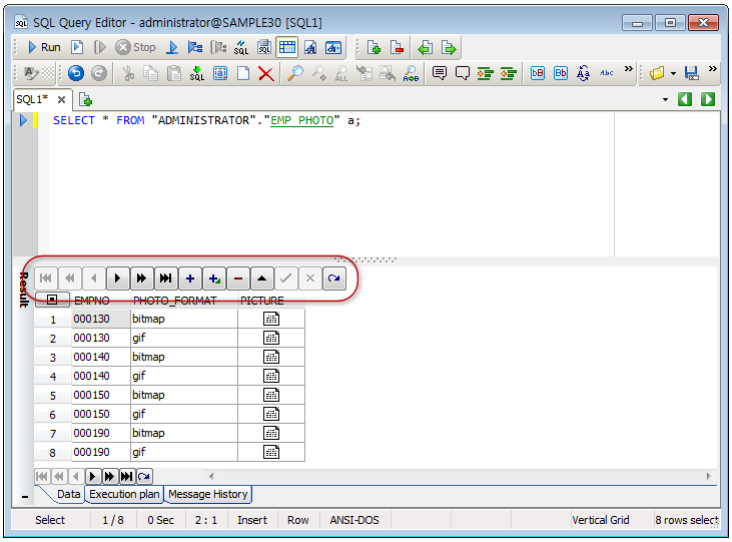
3.4.5. Run Current SQL & Modify

You can run created SQL in SQL Query Editor and modify data in grid.

[Run Current SQL and Modify] is available to run for one Select query. Primary Key column should be included to select script to run.

Select **[Query] > [Run Current SQL and Modify]** on t he main menu or press the **[** **]** button or **< F6 >** key after creating Select query statement.

[Navigation is displayed at grid of **[Data]** tab as modification mode and you can insert/modify/delete by selecting grid.



3.4.6. Execution plan of current SQL

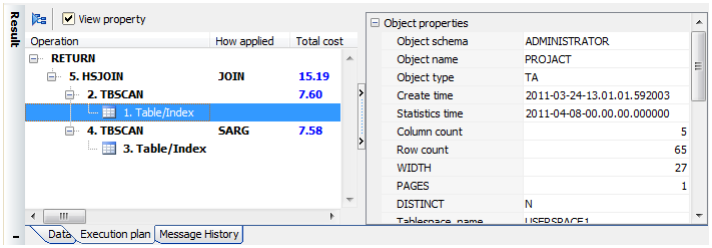
Executes Execution Plan for created SQL in SQL Query Editor.

Results will be shown as tree type after executing 'Execution Plan' command, which saves execution plan to plan table.

Plan table must exist to execute Execution Plan.

Click **[Query] > [Execution plan of current SQL < F7 >]** on the main menu or the **[** **]** button.

Results can be checked at **[Execution plan]** tab.

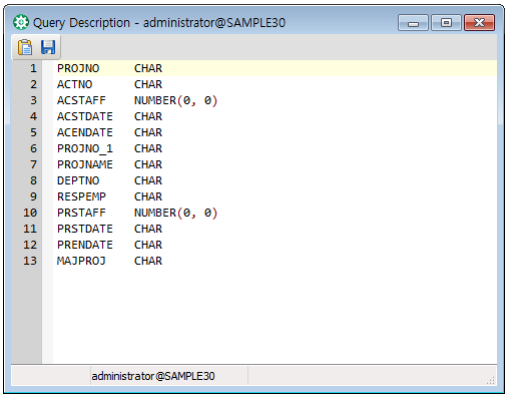


If you select the Object Name after checking the **[View execution plan Object]**, information about that object will be displayed at the right.

3.4.7. Query Description

You can preview columns by using **Query Description** feature before executing a query manually if it is created as subquery type, joined query of several tables or took long time to execute.

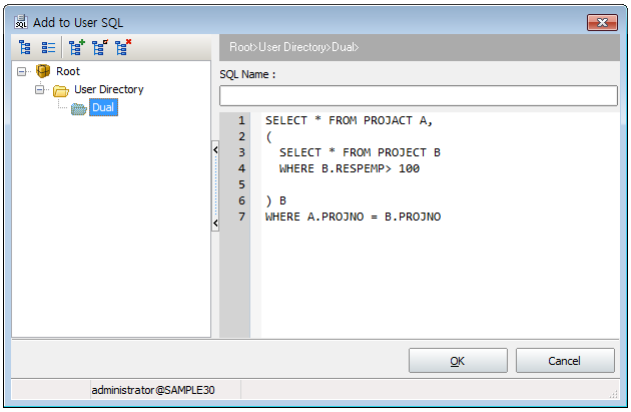
1. Create SQL in SQL Query Editor.
2. Select **[Query] > [Query Description < Shift+F4 >]** on the main menu or press the **[** **]** button.
3. Column information of SQL will be shown.



3.4.8. Add to User SQL

Saves created SQL in SQL Query Editor to **User Directory** of **[SQL HISTORY… < F8 >]**.

Select **[Query] > [Add to User SQL < Shift+F8 >]** on the main menu.



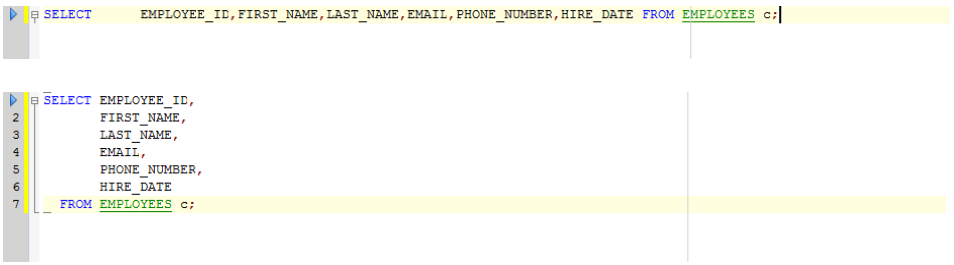
If you click the **[OK]** button after selecting SQL name and group, it will be saved.

Saved user SQL can be checked in **[SQL HISTORY… < F8 >]**.

3.4.9. SQL Align

Aligns created SQL in SQL Query Editor automatically.

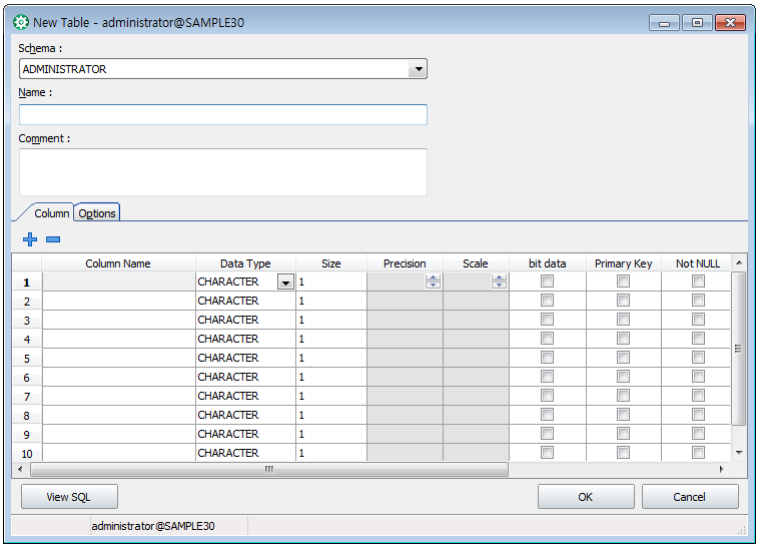
Press **[Query] > [SQL Align < F9 >]** on the main menu or the **[ ]** after creating SQL.



3.5. Create

3.5.1. New Table

Creates table on database. Select **[Create] > [Table]** on the main menu.



- **Column Tab**

Creates table by inputting column information to create through row grid.

1. Select an owner **Schema** of table.
2. Input a **Table name**.
3. Input **Comment** of table.
4. Input a name at **Column Name** in row grid.
5. Select a data type of column at **Data Type**.
6. Input size of column at **Size** input box.
7. Check **Primary Key** in case of primary key. If you check **Primary Key**, **Not NULL** will be checked automatically.
8. Input default value at **Default** input box.
9. Input comment of field at **Comment** input box.
10. Repeat above steps as much as to add rows.
11. Check SQL by clicking the **[View SQL]** button.
12. Table will be created after clicking the **[OK]** button.

※ Addable input boxes of 10 are displayed as default and if you press the **<+>** key, input box will be added.

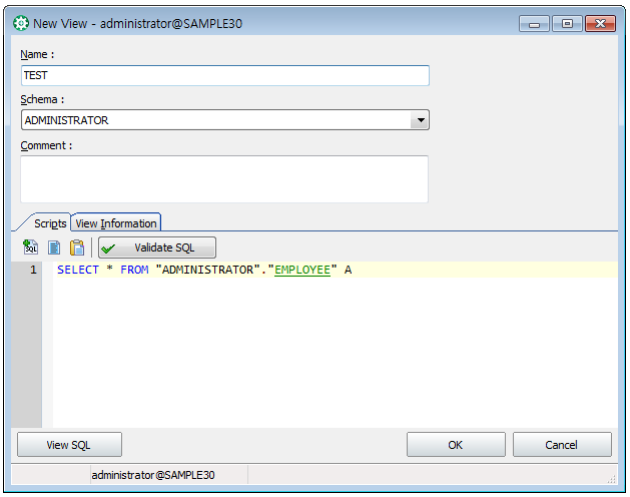
- **Options Tab**

You can specify options, such as Storage, Data Capture, Compression, Logging and so on.

3.5.2. New View

Creates new view on database. To create view, **CREATE VIEW** privilege is necessary.

Select **[Create] > [View ...]** on the main menu.

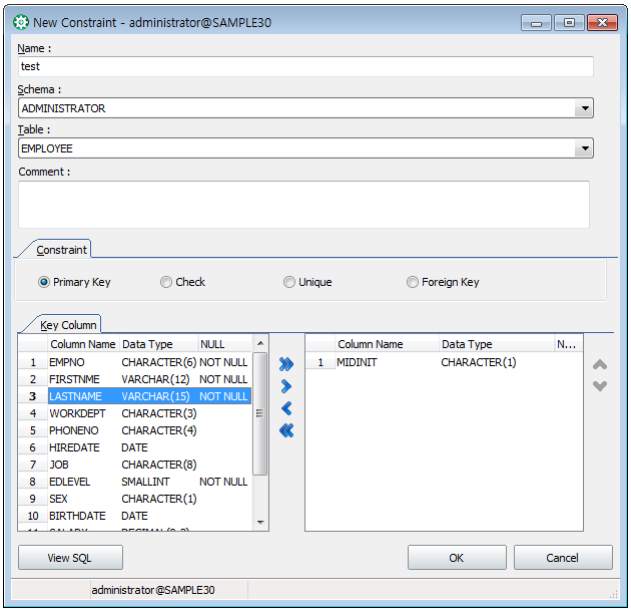


1. Input a **Name**.
2. Select a **Schema to own**.
3. Input SQL to use as view in the **[Scripts]** tab.
4. Click the **[Validate SQL]** button to make sure that SQL is right.
5. Move to the **[View Information]** tab and add column of view by using the **[Auto Add]** button or the **[Add]** b utton.
6. Specify options, such as Replace the view if exists, WITH ROW MOVEMENT and With Check Option.
   - **Replace the view if exists** : If view exists already, overwrite it.
   - **WITH ROW MOVEMENT** : Specify that updated rows move to the right default table, even if the check constraint of the default table is violated**.**
   - **With Check Option** : Specify the restriction, that all rows, inserted or updated through view, must follow the definition of the view.
7. Check SQL by clicking the **[View SQL]** button.
8. Click the **[OK]** button.

3.5.3. New Constraint

Creates primary, check, unique or foreign key on database.

Select **[Create] > [Constraint...]** on the main menu.



- **Primary Key(Primary Key)**

1. Input **Name** of constraint.
2. Select a schema and then select a table to create primary key.
3. Select the **Primary Key** button in **[Constraint]** tab.
4. Select columns to use as primary key at left list in **[Key Column]** tab and then select the [ ] (Add All) or the [ ] (Add) to move to right list.
5. Check SQL by clicking the **[View SQL]** button.
6. Click the **[OK]** button.

- **Check**

1. Input **Name** of constraint.
2. Select a schema and then select a table to create check.
3. Select the **Check** button in **[Constraint]** tab.
4. Input Check Constraint Condition in editor.
5. Check SQL by clicking the **[View SQL]** button.
6. Click the **[OK]** button.

- **Unique**

1. Input **Name** of constraint.
2. Select a schema and then select a table to create unique.
3. Select the **Unique** button in **[Constraint]** tab.

4. Select columns to use as unique key at left list in **[Key Column]** tab and then select the [ ] (Add All) or the [ ] (Add) to move to right list.
5. Check SQL by clicking the **[View SQL]** button.
6. Click the **[OK]** button.

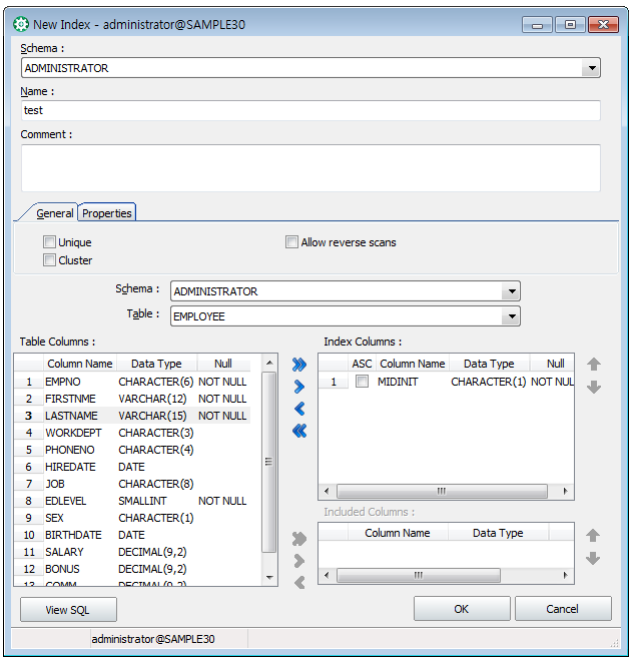
- **Foreign Key**

1. Input **Name** of constraint.
2. Select a schema and then select a table to create foreign key.
3. Select the **Foreign Key** button in **[Constraint]** tab.
4. Select a table to refer in schema and table at right.
5. Select source field at left list and reference filed at right list.
   ※ Just, data types of both fields should be same.
6. Click the [ ] button to move to down list.
7. Repeat 5~6 steps.
8. Check SQL by clicking the **[View SQL]** button.
9. Click the **[OK]** button.

3.5.4. New Index

Creates index of table or cluster on database.

Select **[Create] > [Index …]** on the main menu.



- **General Tab**

1. Specify the **Schema**.
2. Input a name of creating index at **Name**.
3. Move to **[General]** tab. Select **Table**.
4. Select a **Schema** and **Table** to create index.
5. Select columns to use as index at **Table Columns** list and then select the **[** **]** (Add All) or the **[** **]** (Add) to move to right list.
6. Check SQL by clicking the **[View SQL]** button.
7. Click the **[OK]** button.

※ **Unique :** If ON table-name is specified, UNIQUE prevents the table from containing two or more rows with the same value of the index key. The uniqueness is enforced at the end of the SQL statement that updates rows or inserts new rows.

※ **Cluster :** Specifies that the index is the clustering index of the table.

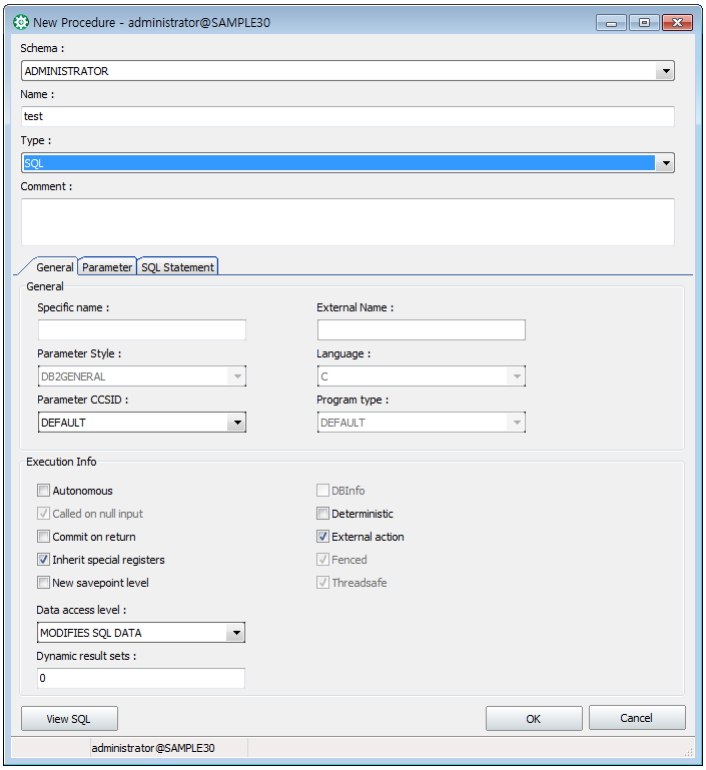
※ **Allow reverse scans** : Specifies that an index can support both forward and reverse scans; that is, scanning of the index in the order that was defined at index creation time, and scanning in the opposite order.

- **Properties**

Specify properties, such as Tablespace, Allocated Space, Page Split, Statistics and so on.

3.5.5. New Procedures

Creates procedures. Select **[Create] > [Procedures...]** on the main menu.



- **General Tab**

1. Specify a **Schema**.
2. Input a name of creating procedure at **Name**.
3. Select a **Type**.
4. Specify each items at the **[General]** tab.
5. Check SQL by clicking the **[View SQL]** button.
6. Click the **[OK]** button.

- **Specific Name** : Provides a unique name for the instance of the procedure that is being defined. This specific name can be used when altering, dropping, or commenting on the procedure.
- **Parameter Style** : This clause is used to specify the conventions used for passing parameters to and returning the value from procedures.
  - **DB2GENERAL** : This means that the procedure will use a parameter passing convention that is defined for use with Java methods. This can only be specified when LANGUAGE JAVA is used.
  - **DB2SQL** : In addition to the parameters on the CALL statement, the following arguments are passed to the procedure.

    A vector containing a null indicator for each parameter on the CALL statement

    The SQLSTATE to be returned to DB2.

    The qualified name of the stored procedure.

    The specific name of the stored procedure.

    The SQL diagnostic string to be returned to DB2
  - **GENERAL** : This means that the procedure will use a parameter passing mechanism by which the procedure

receives the parameters specified on the CALL.

- **GENERAL WITH NULLS** : In addition to the parameters on the CALL statement specified under GENERAL, another argument is passed to the procedure.
- **JAVA** : This means that the procedure will use a parameter passing convention that conforms to the Java language and SQLJ Routines specification.
- **SQL** : In addition to the parameters on the CALL statement, the following arguments are passed to the procedure.

  A null indicator for each parameter on the CALL statement

  The SQLSTATE to be returned to DB2

  The qualified name of the procedure

  The specific name of the procedure

  The SQL diagnostic string to be returned to DB2

- **Parameter CCSID** : Specifies the encoding scheme to use for all string data passed into and out of the procedure.
  - **ASCII** : Specifies that string data is encoded in the database code page.
  - **UNICODE** : Specifies that string data is encoded in Unicode.
- **Program type** : Specifies whether the procedure expects parameters in the style of a main routine or a sub routine. The default is SUB.
- **Language** : This mandatory clause is used to specify the language interface convention to which the procedure body is written.
  - **C** : This means the database manager will call the procedure as if it were a C procedure.
  - **JAVA** : This means the database manager will call the procedure as a method in a Java class.
- **NEW SAVEPOINT LEVEL**: Specifies whether or not this procedure establishes a new savepoint level for savepoint names and effects. OLD SAVEPOINT LEVEL is the default behavior.
- **DBInfo**: Specifies whether specific information known by DB2 is passed to the procedure when it is invoked as an additional invocation-time argument (DBINFO) or not (NO DBINFO). NO DBINFO is the default.
- **Deterministic** : This clause specifies whether the procedure always returns the same results for given argument values (DETERMINISTIC) or whether the procedure depends on some state values that affect the results (NOT DETERMINISTIC).
- **External Action** : Specifies whether the procedure takes some action that changes the state of an object not managed by the database manager (EXTERNAL ACTION), or not (NO EXTERNAL ACTION). The default is EXTERNAL ACTION.
- **Fenced** : This clause specifies whether the procedure is considered "safe" to run in the database manager operating environment's process or address space (NOT FENCED), or not (FENCED).
- **Threadsafe** : Specifies whether the procedure is considered safe to run in the same process as other routines (THREADSAFE), or not (NOT THREADSAFE).
- **Data access level** : Indicates whether the procedure issues any SQL statements and, if so, what type.
  - **NO SQL** : Indicates that the procedure cannot execute any SQL statements.
  - **CONTAINS SQL** : Indicates that SQL statements that neither read nor modify SQL data can be executed by the procedure.
  - **READS SQL DATA** : Indicates that some SQL statements that do not modify SQL data can be included in the procedure.

- **MODIFIES SQL DATA** : Indicates that the procedure can execute any SQL statement except statements that are not supported in procedures.

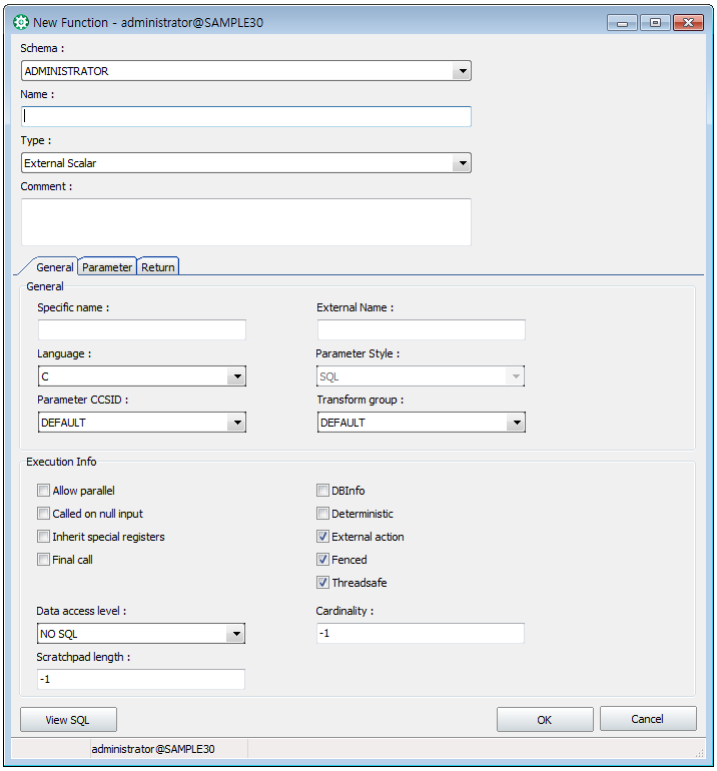
- **Dynamic Result Sets** : Indicates the estimated upper bound of returned result sets for the procedure.

- **Parameter Tab**

- **IN** : Identifies the parameter as an input parameter to the procedure. Any changes made to the parameter within the procedure are not available to the calling SQL application when control is returned. The default is IN.
- **OUT** : Identifies the parameter as an output parameter for the procedure.
- **INOUT** : Identifies the parameter as both an input and output parameter for the procedure.

3.5.6. New Function

Creates functions. Select **[Create] > [Function...]** on the main menu.



1. Specify a **Schema**.
2. Input a name of creating function at **Name**.
3. Select a **Type**.
4. Specify each items at the **[General]** tab.
5. Check SQL by clicking the **[View SQL]** button.
6. Click the **[OK]** button.

- **Function Type**

- **External Scalar** : The CREATE FUNCTION (External Scalar) statement is used to register a user-defined external scalar function at the current server. A scalar function returns a single value each time it is invoked, and is in general valid wherever an SQL expression is valid.
- **External Table** : The CREATE FUNCTION (External Table) statement is used to register a user-defined external table function at the current server.
  A table function can be used in the FROM clause of a SELECT, and returns a table to the SELECT by returning one row at a time.
- **SQL Scalar/SQL Table/SQL Row** : The CREATE FUNCTION (SQL scalar, table, or row) statement is used to define a user-defined SQL scalar, table, or row function. A scalar function returns a single value each time it is invoked, and is generally valid wherever an SQL expression is valid. A table function can be used in a FROM clause and returns a table. A row function can be used as a transform function and returns a row.
- **Sourced/Template** : The CREATE FUNCTION (Sourced or Template) statement is used to.
  - Register a user-defined function, based on another existing scalar or aggregate function, at the current server.
    Register a function template with an application server that is designated as a federated server. A function template is a partial function that contains no executable code. The user creates it for the purpose of mapping it to a data source function.

● General Tab

- **Specific Name** : Provides a unique name for the instance of the function that is being defined. This specific name can be used when sourcing on this function, dropping the function, or commenting on the function.
- **External Name** : This clause indicates that the CREATE FUNCTION statement is being used to register a new function based on code written in an external programming language and adhering to the documented linkage conventions and interface.
- **Language** : This mandatory clause is used to specify the language interface convention to which the user-defined function body is written.
  - **C** : This means the database manager will call the user-defined function as if it were a C function.
  - **JAVA** : This means the database manager will call the user-defined function as a method in a Java class.
  - **CLR** : This means the database manager will call the user-defined function as a method in a .NET class.
  - **OLE** : This means the database manager will call the user-defined function as if it were a method exposed by an OLE automation object.
- **Parameter CCSID** : Specifies the encoding scheme to use for all string data passed into and out of the function.
  - **ASCII** : Specifies that string data is encoded in the database code page.
  - **UNICODE** : Specifies that character data is in UTF-8, and that graphic data is in UCS-2.
- **Parameter Style** : This clause is used to specify the conventions used for passing parameters to and returning the value from functions.
  - **DB2GENERAL** : Used to specify the conventions for passing parameters to and returning the value from external functions that are defined as a method in a Java class.
  - **SQL** : Used to specify the conventions for passing parameters to and returning the value from external functions that conform to C language calling and linkage conventions, methods exposed by OLE automation objects, or public static methods of a .NET object.

- **Transform Group** : Indicates the transform group to be used for user-defined structured type transformations when invoking the function. A transform is required if the function definition includes a user-defined structured type as either a parameter or returns data type. If this clause is not specified, the default group name DB2_FUNCTION is used.
- **Allow Parallel** : Specifies whether or not, for a single reference to the function, the invocation of the function is to be parallelized.
- **Called on null input** : If CALLED ON NULL INPUT is specified, then regardless of whether any arguments are null, the user-defined function is called. It can return a null value or a normal (non-null) value.
- **Inherit special registers** : This optional clause specifies that updatable special registers in the function will inherit their initial values from the environment of the invoking statement. For a function invoked in the select-statement of a cursor, the initial values are inherited from the environment when the cursor is opened.
- **Final Call** : This optional clause specifies whether a final call (and a separate first call) is to be made to an external function. It also controls when the scratchpad is re-initialized. If NO FINAL CALL is specified, then DB2 can only make three types of calls to the table function: open, fetch and close.
- **DBInfo** : This optional clause specifies whether certain specific information known to DB2 is to be passed to the function as an additional invocation-time argument (DBINFO) or not (NO DBINFO). NO DBINFO is the default.
- **Deterministic** : This optional clause specifies whether the function always returns the same results for given argument values (DETERMINISTIC) or whether the function depends on some state values that affect the results (NOT DETERMINISTIC). That is, a DETERMINISTIC function must always return the same table from successive invocations with identical inputs.
- **External action** : Specifies whether the function takes an action that changes the state of an object that the database manager does not manage.
- **Fenced** : This clause specifies whether or not the function is considered “safe” to run in the database manager operating environment's process or address space (NOT FENCED), or not (FENCED).
- **Threadsafe** : Specifies whether the function is considered safe to run in the same process as other routines (THREADSAFE), or not (NOT THREADSAFE).
- **Data access level** :
  - **NO SQL** : Indicates that the function cannot execute any SQL statements.
  - **CONTAINS SQL** : Indicates that SQL statements that neither read nor modify SQL data can be executed by the function.
  - **READS SQL DATA** : Indicates that SQL statements that do not modify SQL data can be executed by the function.
- **Cardinality** : This optional clause provides an estimate of the expected number of rows to be returned by the function for optimization purposes.
- **Scratchpad length** : This optional clause may be used to specify whether a scratchpad is to be provided for an external function. A scratchpad provides a means for the function to “save state” from one call to the next.

● Return Tab

- **Return Type** : Specifies the data type of the output.
- **Cast From** : This form of the RETURNS clause is used to return a different data type to the invoking state

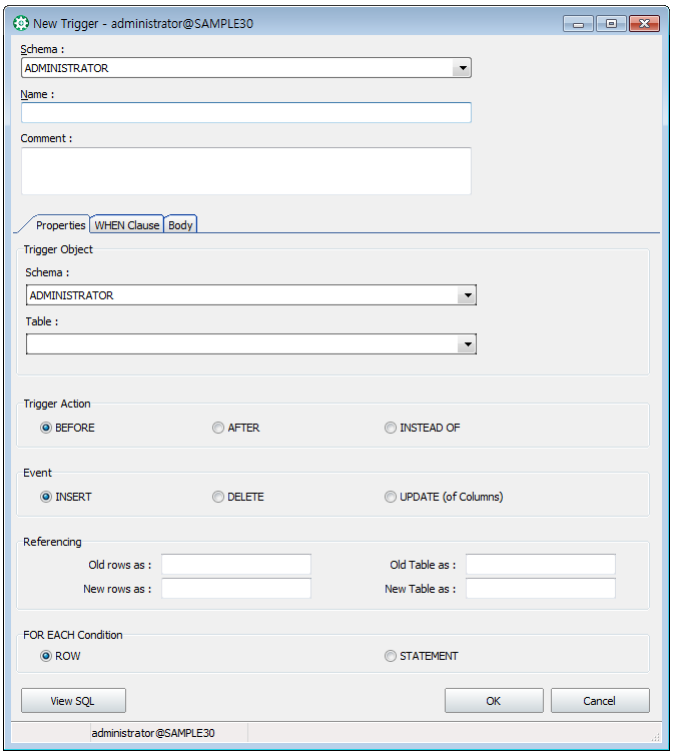
ment from the data type that was returned by the function code.

- **As Locator** : Specifies that a locator to the value of the parameter is passed to the function instead of the actual value.

3.5.7. New Trigger

Creates triggers on the database.

Select **[Create] > [Trigger...]** on the main menu.



1. Specify a **Schema**.
2. Input a name of creating trigger at **Name**.
3. Specify each items at the **[Properties]** tab.
4. Check SQL by clicking the **[View SQL]** button.
5. Click the **[OK]** button.

- **Properties**

- **BEFORE** : Specifies that the associated triggered action is to be applied before any changes caused by the actual update of the subject table are applied to the database. It also specifies that the triggered action of the trigger will not cause other triggers to be activated.
- **AFTER** : Specifies that the associated triggered action is to be applied after the changes caused by the actual update of the subject table are applied to the database.
- **INSTEAD OF** : Specifies that the associated triggered action replaces the action against the subject view. Only one INSTEAD OF trigger is allowed for each kind of operation on a given subject view.
- **INSERT** : Specifies that the triggered action associated with the trigger is to be executed whenever an INSE

RT operation is applied to the subject table or subject view.

- **DELETE** : Specifies that the triggered action associated with the trigger is to be executed whenever a DELETE operation is applied to the subject table or subject view.
- **UPDATE** : Specifies that the triggered action associated with the trigger is to be executed whenever an UPDATE operation is applied to the subject table or subject view, subject to the columns specified or implied.
- **OLD ROWS AS** : Specifies a correlation name which identifies the row state prior to the triggering SQL operation.
- **NEW ROWS AS** : Specifies a correlation name which identifies the row state as modified by the triggering SQL operation and by any SET statement in a BEFORE trigger that has already executed.
- **OLD_TABLE AS** : Specifies a temporary table name which identifies the set of affected rows prior to the triggering SQL operation.
- **NEW_TABLE AS** : Specifies a temporary table name which identifies the affected rows as modified by the triggering SQL operation and by any SET statement in a BEFORE trigger that has already executed.
- **FOR EACH ROW** : Specifies that the triggered action is to be applied once for each row of the subject table or subject view that is affected by the triggering SQL operation.
- **FOR EACH STATEMENT** : Specifies that the triggered action is to be applied only once for the whole statement.

- **When Clause**

Input conditions to execute trigger.

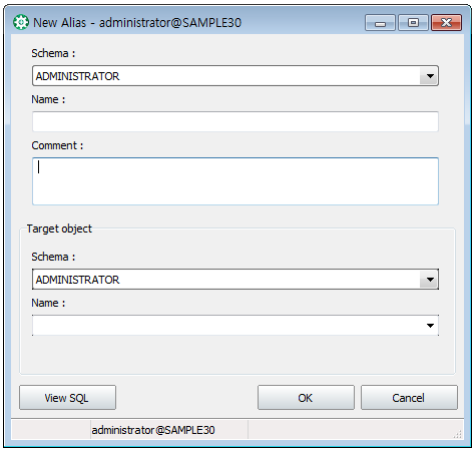
- **Body**

Describe procedure to call when executes trigger.

3.5.8. New Alias

Creates aliases for table, view, alias or other aliases.

Select **[Create] > [Alias...]** on the main menu.



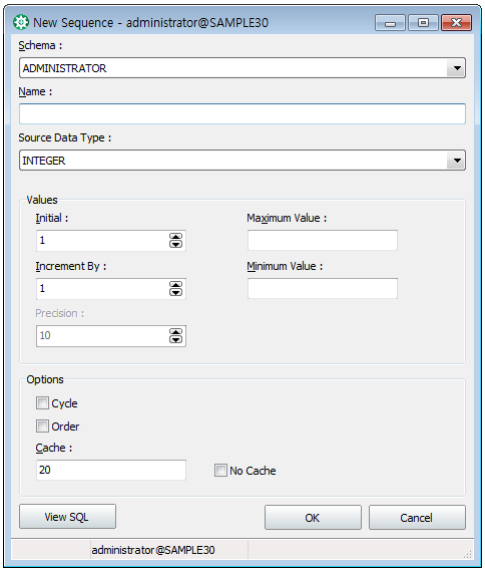
1. Specify a **Schema**.

2. Input a name of creating alia at **Name**.
3. Specify a schema and table of target object.
4. Check SQL by clicking the **[View SQL]** button.
5. Click the **[OK]** button to apply the alias.

3.5.9. New Sequences

Defines sequences on the application program server.

Select **[Create] > [Sequences...]** on the main menu.



1. Specify a **Schema**.
2. Input a name of creating sequence at **Name**.
3. Specify a **Source Data Type**.
4. Input the **Value** option.
5. Select **Options**.
6. Check SQL by clicking the **[View SQL]** button.
7. Click the **[OK]** button.

※ **Source Data Type** : Data type, used to sequence value

- **Value**

- **Initial** : Input initial value of sequence.
- **Minimum Value** : Input minimum value of sequence.
- **Maximum Value** : Input maximum value of sequence.
- **Increment By** : Input increment value of sequence.

- **Options**

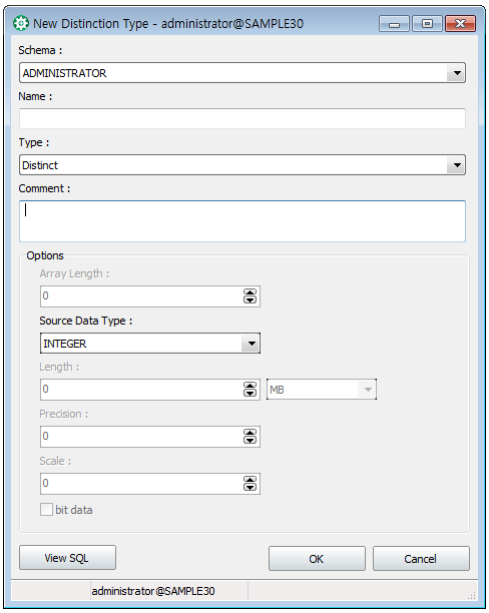
- **Cycle** : Specify to create sequence value continuously after reaching maximum or minimum value.
- **Order** : Select whether to allocate number orderly when sequence create number.
- **Cache** : Specify whether to preserve some of pre-allocated values to memory for faster access.
- This is option for performance and adjustment.

3.5.10. New Distinct type

Creates changeable and new Varying Array or Nested Table on the database.

To create type, **CREATE TYPE** or **CREATE ANY TYPE** privilege is necessary.

Select **[Create] > [Distinct type...]** on the main menu.

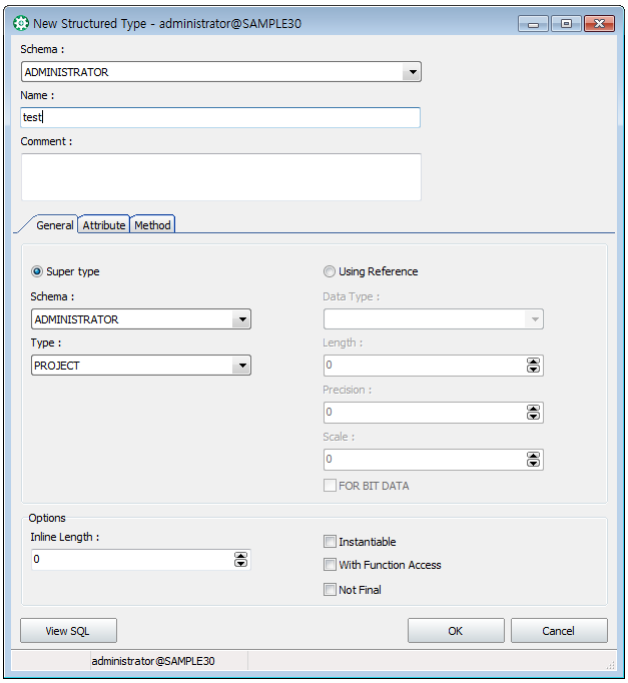


1. Specify a **Schema**.
2. Input a name of creating Distinct type at **Name**.
3. Select a **Type**.
4. Specify each items.
5. Check SQL by clicking the **[View SQL]** button.
6. Click the **[OK]** button.

3.5.11. New Structured type

Defines user-defined **Structured type**.

Select **[Create] > [Structured type...]** on the main menu.



1. Specify a **Schema**.
2. Input a name of creating structured type at **Name**.
3. Select a **Type**.
4. Specify each items at the **[General]** tab.
5. Check SQL by clicking the **[View SQL]** button.
6. Click the **[OK]** button.

● General

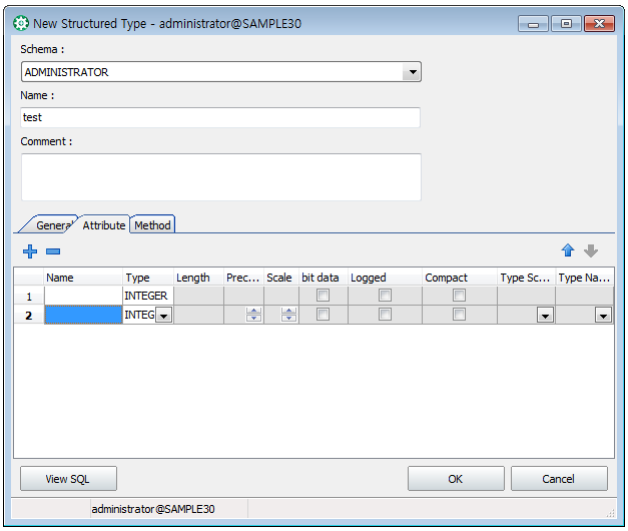
- **SuperType** : Specifies a supertype. A user-defined structured type may include zero or more attributes. A structured type may be a subtype allowing attributes to be inherited from a supertype.
- **Ref Using** : Defines the built-in data type used as the representation (underlying data type) for the referenc e type of this structured type and all its subtypes.
- **Inline Length** : This option indicates the maximum size (in bytes) of a structured type column instance to s tore inline with the rest of the values in the row of a table. Instances of a structured type or its subtypes, that are larger than the specified inline length, are stored separately from the base table row, similar to t he way that LOB values are handled.
- **Instantiable** : Determines whether an instance of the structured type can be created.
  Implications of not instantiable structured types are.
  - no constructor function is generated for a non-instantiable type.
  - a non-instantiable type cannot be used as the type of a table or view.
  - a non-instantiable type can be used as the type of a column.
    (only null values or instances of instantiable subtypes can be inserted into the column).
- **With Function Access** : Indicates that all methods of this type and its subtypes, including methods created in the future, can be accessed using functional notation.
- **Not Final** : Indicates that the structured type may be used as a supertype.

- **Attribute**

1. Move to the **[Attribute]** tab.

2. When click the [ ] button, a row will be added.

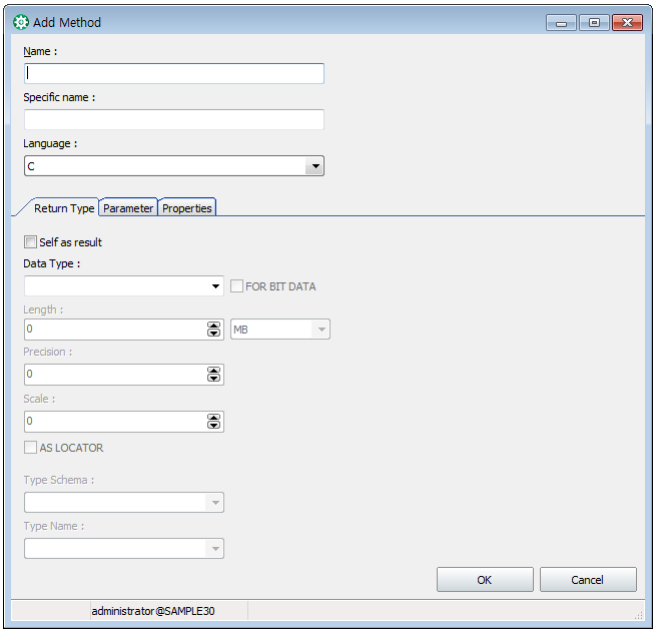
3. Input **Name, Type** and so on, click the **[OK]** button.



- **Method**

1. Move to the **[Method]** tab.

2. When click the [ ] button, **[Add Method]** window will be opened.



3. Specify **[Return Type], [Parameter], [Properties]** and so on.

- **Access Data** :
  - **NO SQL** : Represents that the method cannot execute SQL.
  - **CONTAINS SQL** : Represents that SQL, which didn't read or modify SQL data, cannot execute by the method.

- **READS SQL DATA** : Represents that SQL, which didn't modify SQL data, cannot be included to the method.

- **Scratchpad** : If you specify the SCRATCHPAD,
  additional factor is sent to the external method, which specify the scratch pad, whenever it calls user-defined method.

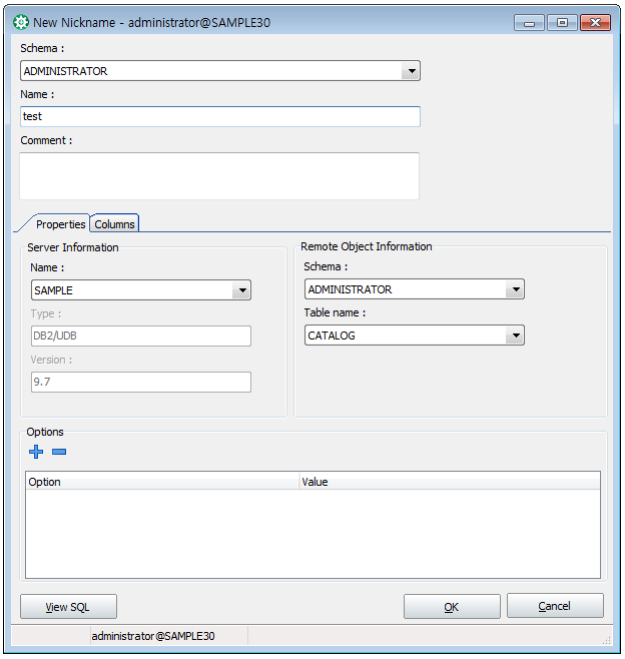
4. Click the **[OK]** button.

3.5.12. New Nickname

Defines a **New Nickname**.

Select **[Create] > [Nickname...]** on the main menu.

If you set an alias, you can control remote objects in remote.



To call server information, you must register Wrapper after turning on Federated System.

After that Server Information and Remote Object Information can be called.

DB2 server is called Federated Server in Federated System.

You can configure DB2 Instances as much as wanted count to operate as Federated Server.

Moreover, you can use the existing DB2 Instance as Federated Server or add separately for Federated System.

Above configuration can be set by using DB2 Control Center. Or register Wrapper by using Federated Server with CREATE WRAPPER statement.

Wrapper has mechanism that Federated Server interacts with specific type of data source.

1. Specify a **Schema**.
2. Input a name of creating alias at **Name**.
3. Select **[Server Information]**.
4. Select **[Remote Object Information]**.
5. Check SQL by clicking the **[View SQL]** button.
6. Click the **[OK]** button.

3.6. Tools

3.6.1. Reports

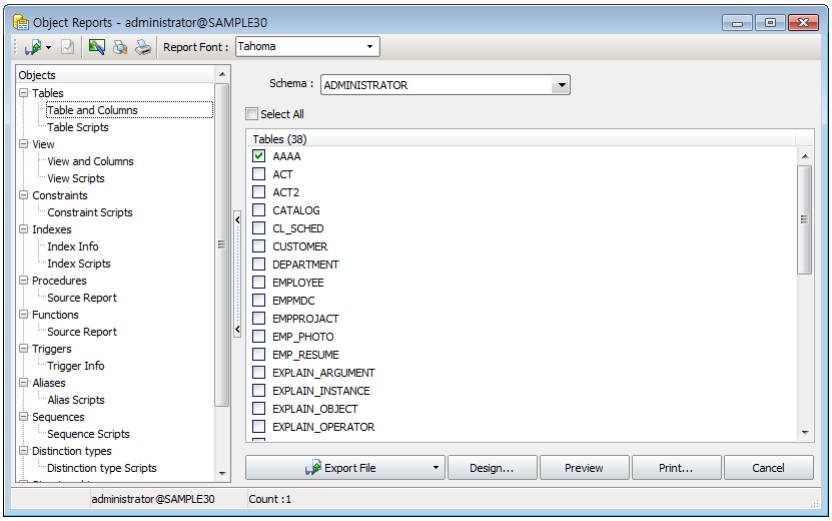
If you use Object Report Tool, you can design and print out various reports by yourself.

You can handle report writing easily and fast because report can be designed and printed out in template format.

Object report template files are stored in Report folder in .fr3 file format.

You can select report type at left panel and set printout options at right panel in object report tool.

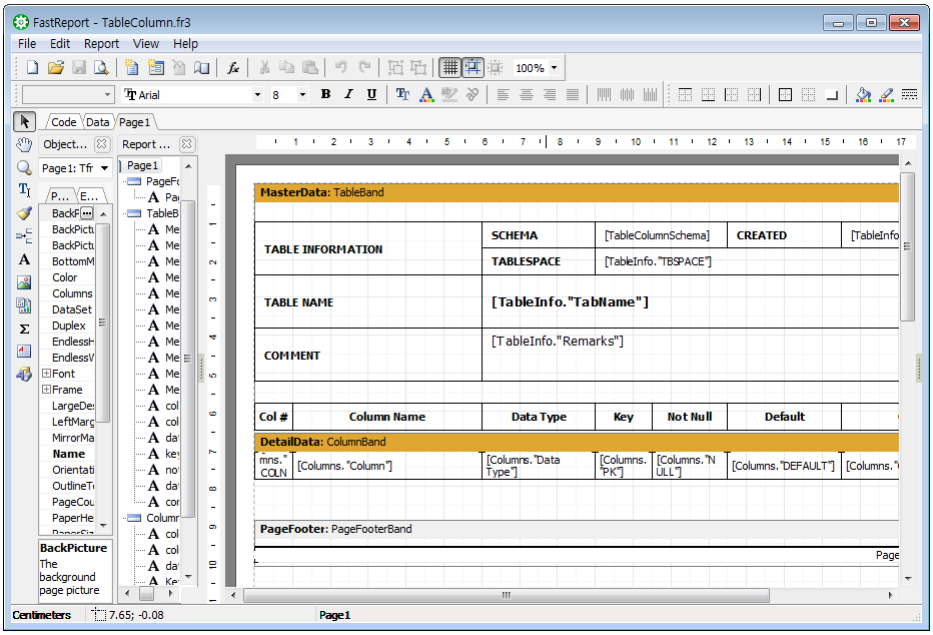
Select **[Tools] > [Reports] > [Object Reports]** on the main menu or click the **[** **]** button.



1. Select a type of report to print out at left tree.
2. Select a schema and object to print out at right panel.
3. Select font to print out at the **Report Font.**
4. If you click the **[Design ...]** button, you can design reports.
5. If you click the **[Preview]** button, you can it in advance.
6. If you click the **[Print...]** button, report will be printed out.

※ If you click the **[Export File]** button, you can select format to print out reports.
(PDF, EXCEL, RTF, BMP, JPG, GIF, TEXT, E-Mail, HTML)

- **Design Report**



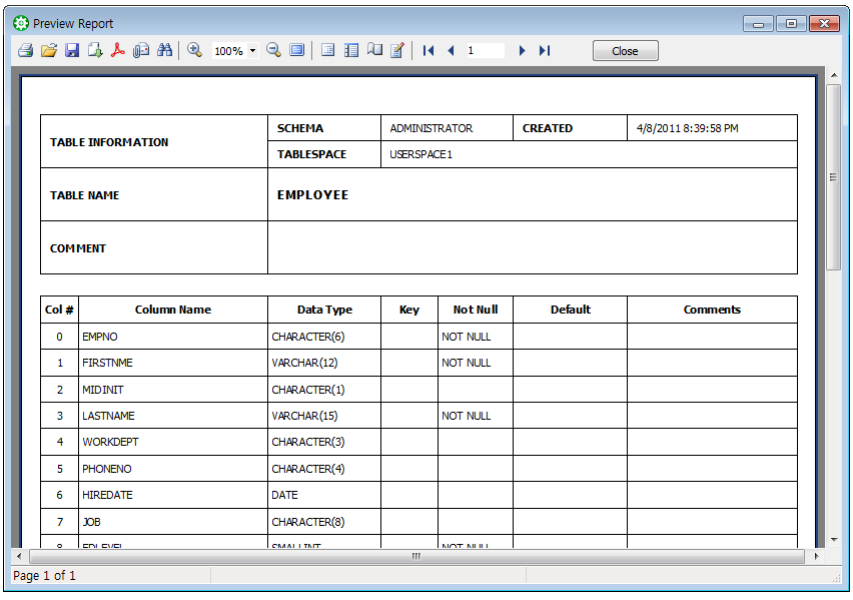
Select a report at left tree.

Click the **[Design]** button.

You can edit design to other formats that you want to print out.

Select the **[File] > [Save]** buttons after editing.

- **Preview Report**



Col #	Column Name	Data Type	Key	Not Null	Default	Comments
0	EMPNO	CHARACTER(6)		NOT NULL		
1	FIRSTNME	VARCHAR(12)		NOT NULL		
2	MIDINIT	CHARACTER(1)				
3	LASTNAME	VARCHAR(15)		NOT NULL		
4	WORKDEPT	CHARACTER(3)				
5	PHONENO	CHARACTER(4)				
6	HIREDATE	DATE				
7	JOB	CHARACTER(8)				

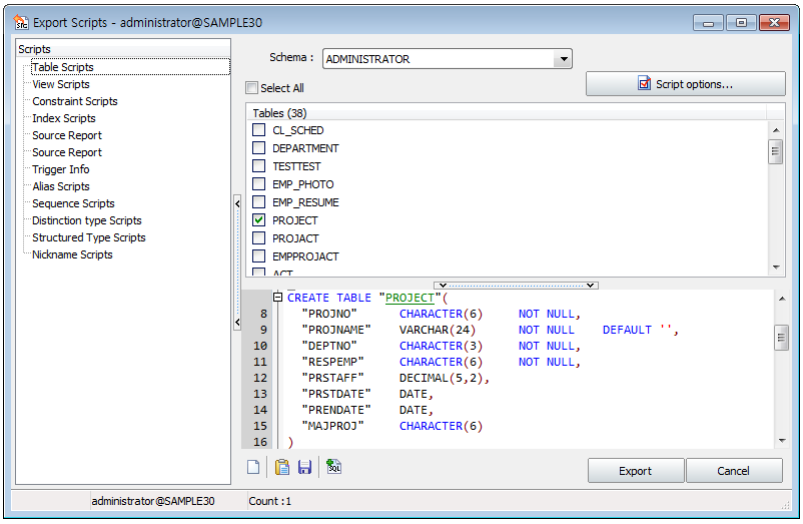
Select a report at left tree.

Click the **[Preview]** button.

3.6.2. Export Scripts

If you use Script Export, you can extract DDL scripts information of several database objects and send them to SQL Query Editor easily.

Select **[Tools] > [Reports] > [Export Scripts]** on the main menu or click the [ ] button.



1. Select an object type to export at left tree.
2. Select a schema at schema drop&down list at the top of the screen.
3. Check an object to export.
4. If you click the **[Export]** button, DDL script of the selected object will send to editor.
5. You can add several DDL scripts to editor by repeating step 1 to 4.
6. If you click the [ ]button, editor will be initialized.
7. If you click the [ ]button, exported script in editor sends to SQL Query Editor.

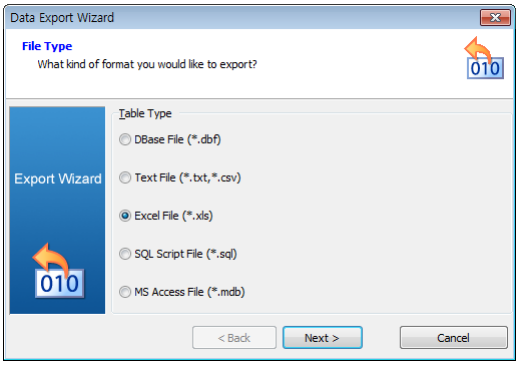
3.6.3. Export Data

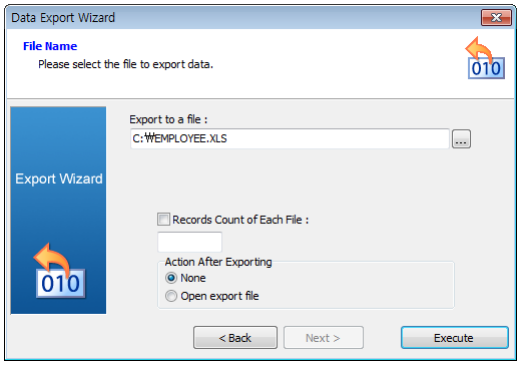
You can export data in table by using Export Data in various formats, such as MS Excel, DBF, Text, CSV File and so on.

Select **[Tools] > [Export Data...]** on the main menu or click the **[** **]** button.

1. Select a table to export, then click the **[Start]** button.

2. Select a file to export after specifying a **File Type**.





If records exceed 6 hundred thousand, a file will be created additionally every 6 hundred thousand by selecting **Records Count of Each File**.

※ Ex) If you input a file name to export as ExportObject.xls, ExportObject001.xls, and ExportObject002.xls files will be created.

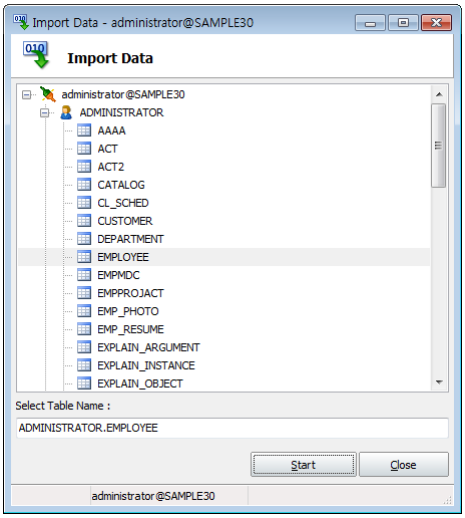
3. Click the **[Run]** button.

3.6.4. Import Data

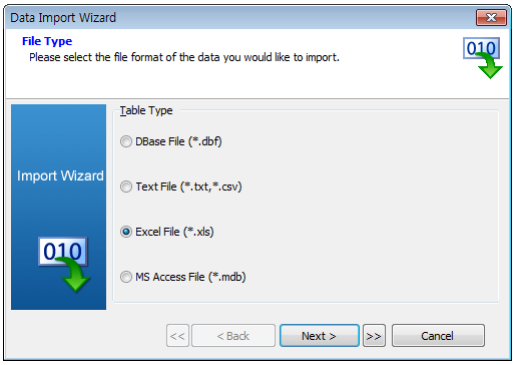
You can import data in various formats files, such as MS Excel, DBF, Text, CSV File and so on, to table by using Import Data.

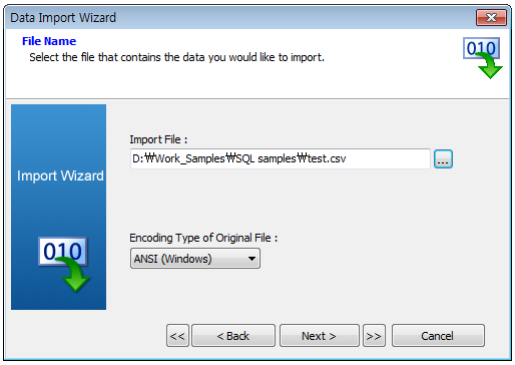
Select **[Tools] > [Import Data...]** on the main menu or click the **[** **]** button.

1. Select a table to import, then click the **[Start]** button.

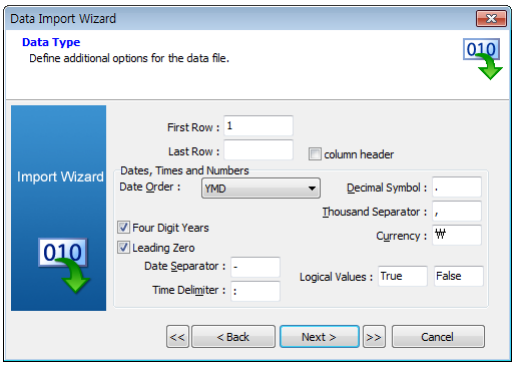


2. Select the original file to import after specifying a file type.





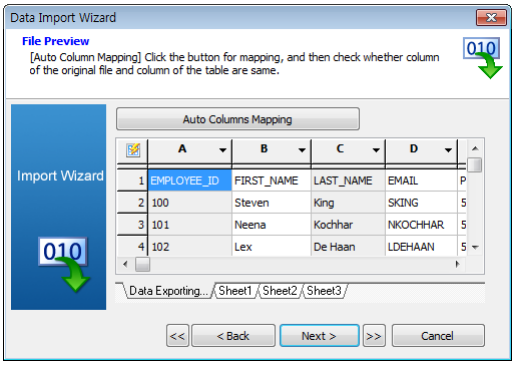
3. Specify data type. Specify Date, Time and Number format.

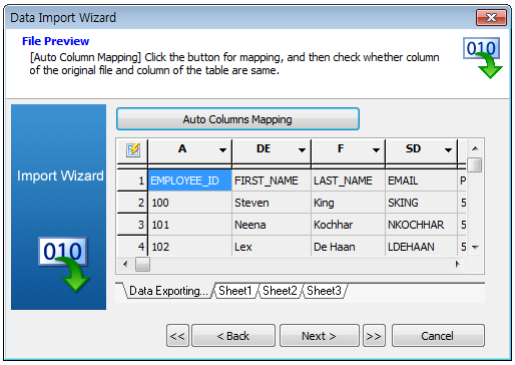


- **First Row** : First Row to import (If there is row name in first row, set as 2.)
- **Last Row** : Last Row to import (Default is blank)

4. Map column of original file onto row of target table.

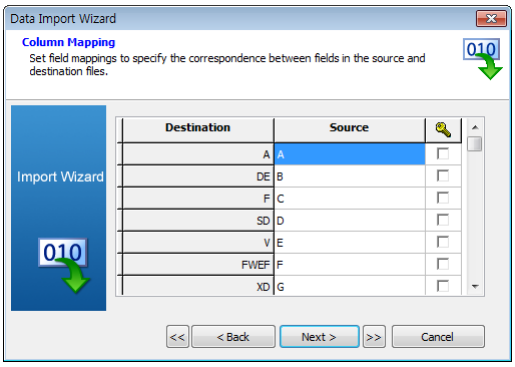
Click the **[Auto Columns Mapping]** button to map automatically, then make sure that each column is mapped correctly.

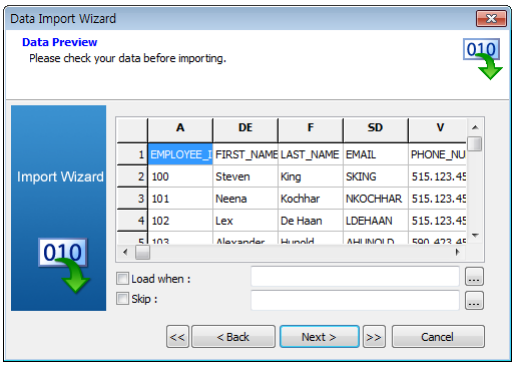




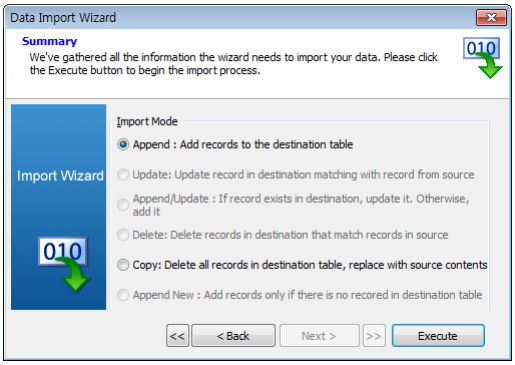
5. Preview data after checking **Column Mapping** between target table and original source file.

   Make sure that mapping between data of original source file and column of target table is done correctly.





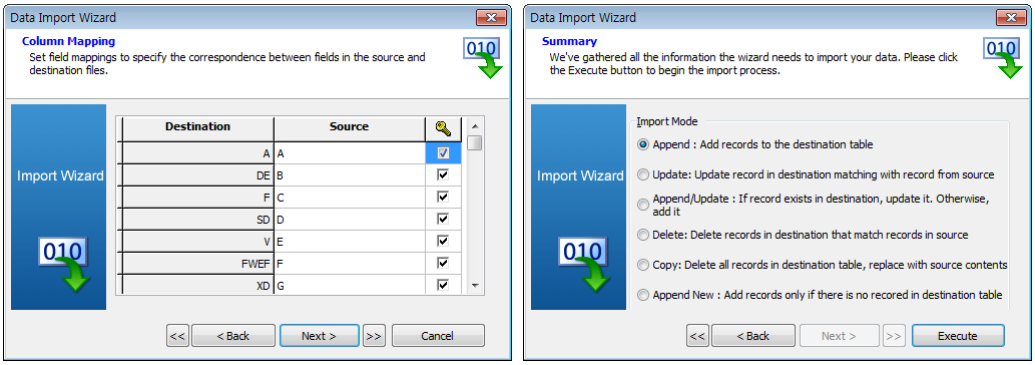
6. Select the **Import Mode** and then click the **[Execute]** button.


Data Import Wizard
Summary
We've gathered all the information the wizard needs to import your data. Please click the Execute button to begin the import process.
Import Wizard
Import Mode
Append : Add records to the destination table
Update: Update record in destination matching with record from source
Append/Update : If record exists in destination, update it. Otherwise, add it
Delete: Delete records in destination that match records in source
Copy: Delete all records in destination table, replace with source contents
Append New : Add records only if there is no recored in destination table
<< < Back Next > >> Execute

- **Add** : Add a record to target table.
- **Copy** : Delete data in target table and adds record to original source file.

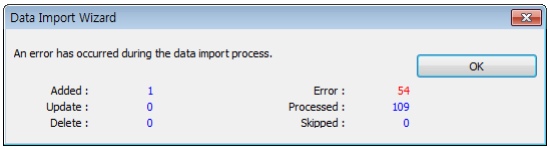
If you set key for row in **Column Mapping** step, you can select mode among **Update**, **Append/Update** and **Delete** mode.

Update item will be activated as follows.


Data Import Wizard
Column Mapping
Set field mappings to specify the correspondence between fields in the source and destination files.
Import Wizard
Destination
Source
A
DE
F
SD
V
FWEF
XD
B
C
D
E
F
G
<< < Back Next > >> Cancel
Data Import Wizard
Summary
We've gathered all the information the wizard needs to import your data. Please click the Execute button to begin the import process.
Import Mode
Append : Add records to the destination table
Update: Update record in destination matching with record from source
Append/Update : If record exists in destination, update it. Otherwise, add it
Delete: Delete records in destination that match records in source
Copy: Delete all records in destination table, replace with source contents
Append New : Add records only if there is no recored in destination table
<< < Back Next > >> Execute

- **Update** : Update corresponding record from original.
- **Append/Update** : Update if record exists in target table or adds if not.
- **Delete** : Delete record if data of original source file exists in target table.
- **Append New** : Add records only if there is no record in target table.

7. Importing result will be shown.



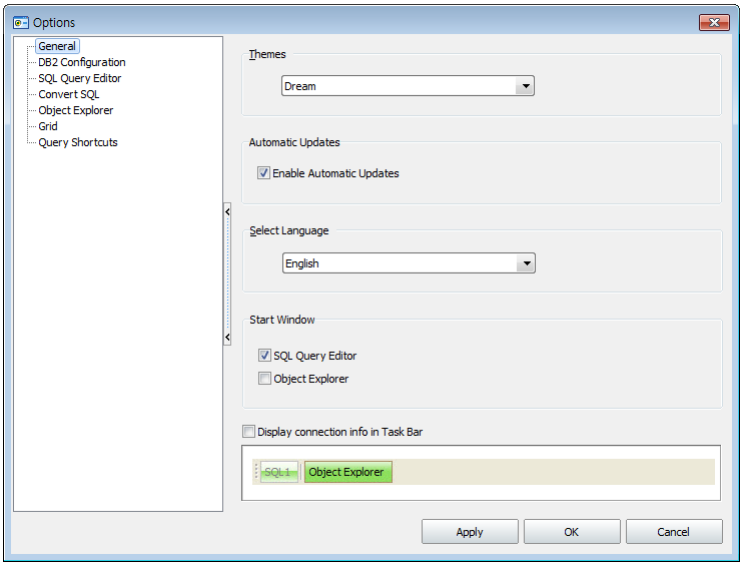
You can check importing result and error record.

3.6.5. Options

Sets options of program.

Select **[Tools] > [Options...]** on the main menu or click the **[ ]** button.

- **[General]**

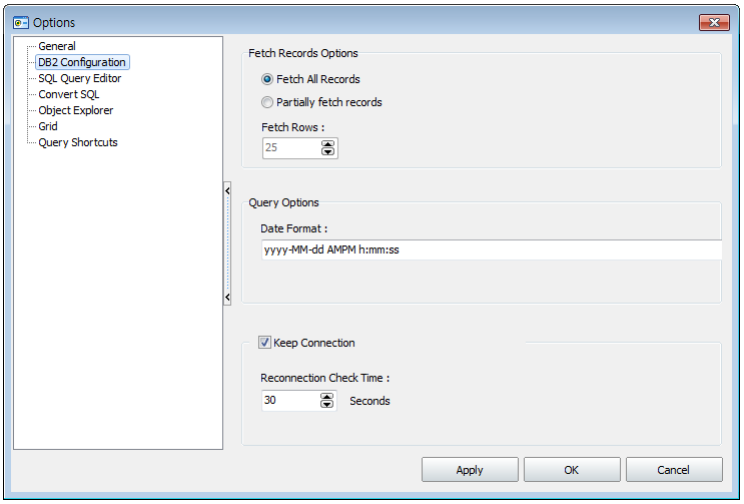


- **Themes** : Select theme of program. Default and Dream are provided.
- **Automatic Updates** : When checked, it checks update automatically whenever program executes.
- **Select Language** : Select a language to use at program. Korean, English, Japanese, Chinese, German, French and Spanish are provided.

- **Start Window** : Select a window to be executed when starts program. Default is SQL Query Editor.
  - **SQL Query Editor** : Execute SQL Query Editor automatically when starts program.
  - **Object Explorer** : Execute Object Explorer automatically when starts program.
- **Display connection info in Task Bar** : Display session ID on task bar.

● [DB2 Configuration]

You can set rows count of query results, execution plan table, user and auto commit.

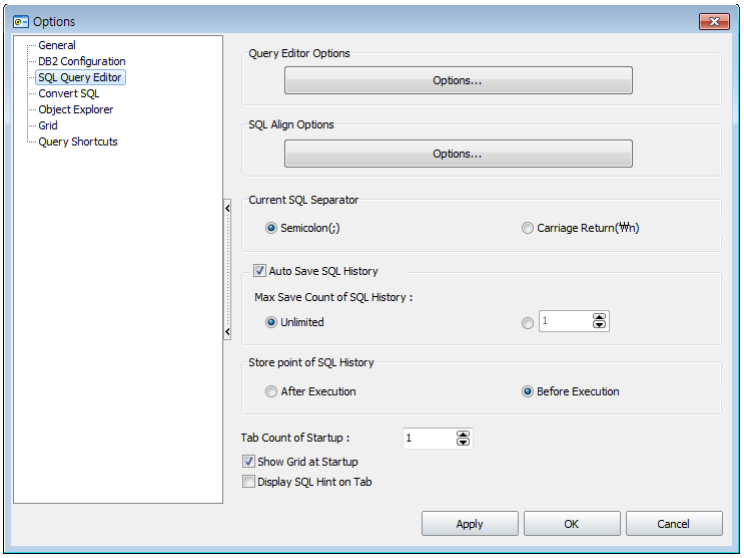


- **Fetch Records Options** :
  - **Fetch All Records** : Display all records in grid after inquiry.
  - **Partially fetch records** : Fetch partially after inquiry. Fetch rows means records count to fetch partially.
- **Query Options** :
  - **Date Format :** Set date type to display in grid.
  - **Execution plan Table Name** : It is table name to use execution plan in SQL Query Editor.

    If you use table of other user, you should input user name or table name.
- **Keep Connection** : When checked, it keeps connection of session continuously.
  - **Reconnection Check Time** : Set automatic reconnection time.

● [SQL QUERY EDITOR]

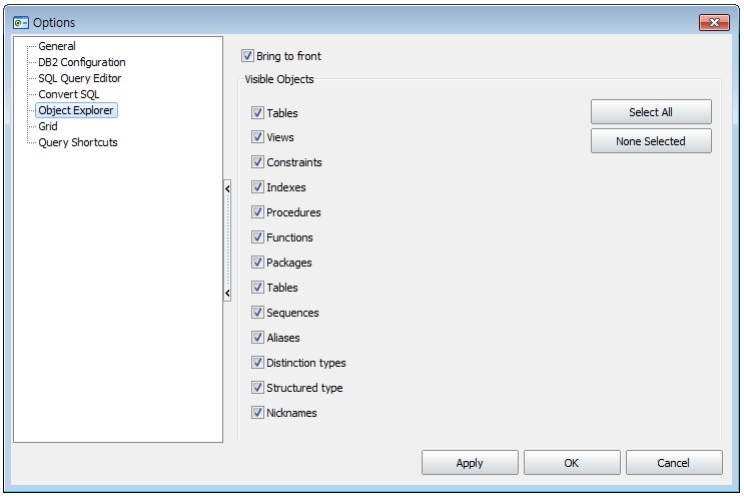
Sets number of start tab, editor options, alignment options, history save options and so on.



- **Query Editor Options** : Set query editor options. If you click the **[Options...]** button, new window will be sh own to set.
- **SQL Align Option** : Set SQL alignment options. If you click the **[Options...]** button, new window will be sho wn to set.
- **Current SQL Separator :** Set which you will use as separator for SQL, semicolon or carriage return.
- **AUTO SAVE SQL HISTORY** : Set options whether to save or not. If you check this, SQL will be saved automa tically.
- **MAX SAVE COUNT OF SQL HISTORY** :
  - **Unlimited** : Save all without limitation.
  - **Count :** Save only inputted count.
- **STORE POINT OF SQL HISTORY** :
  - **After Execution** : Save only succeeded queries.
  - **Before Execution** : Save succeeded or failed queries.
- **Tab Count of Startup** : It is number of tabs, which are created automatically by executing SQL Query Editor.
- **Show Grid at Startup :** When checked, grid will be displayed when executes SQL Query Editor. If it is not ch ecked, only editor will be shown and grid will be shown after executing query.
- **Display SQL Hint on Tab** : When checked, SQL will be shown as hint when mouse over tab.

- **[Object Explorer]**

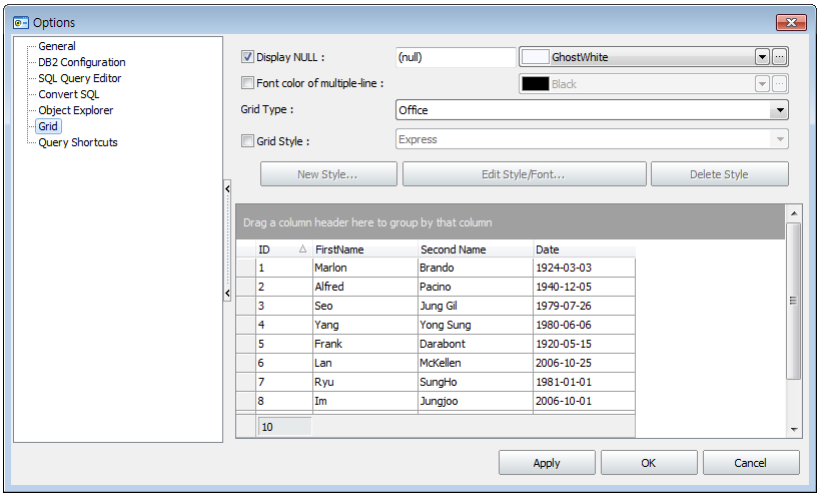
Select objects to display in Object Explorer/Object Panel.



- **Bring to front** : When checked, Schema Browser will be shown on top.

- **[Grid]**

You can change Display Null setting in grid and grid style.

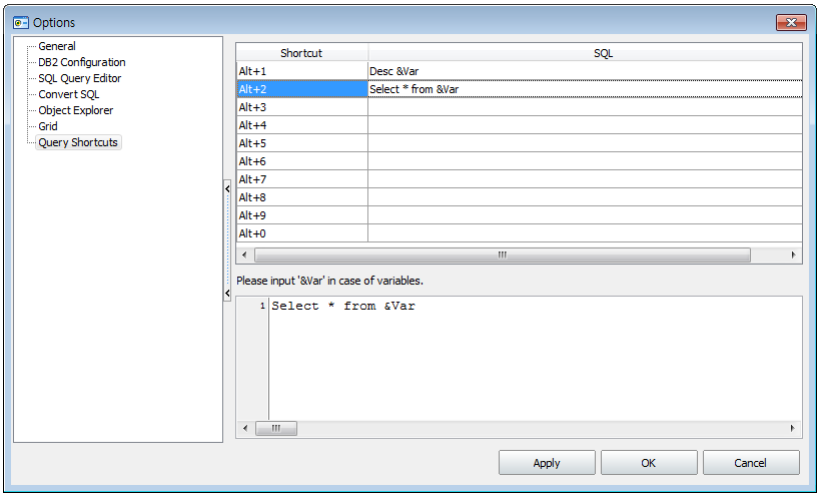


- **Display NULL** : Display NULL value in grid. Set background.
- **Font color of multiple-line** : Set background color when text in grid has carriage return value.

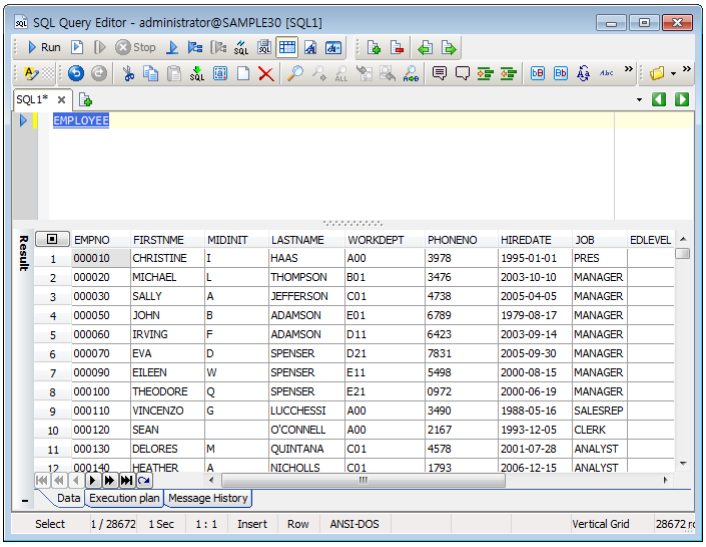
  ※ Only one line of data is displayed in grid, so you cannot know how many lines in grid.
- **Grid Type** : Change grid type. (Standard/Flat/Ultra Flat/Office)
- **Grid Style** : Change grid style. You can change grid style manually.

- **[Query Shortcuts]**

You can set specific commands as shortcut keys, **< Alt+1 ~ 10 >**.



If you select keyword in SQL Query Editor and then press the shortcut key, SQL command will be executed.

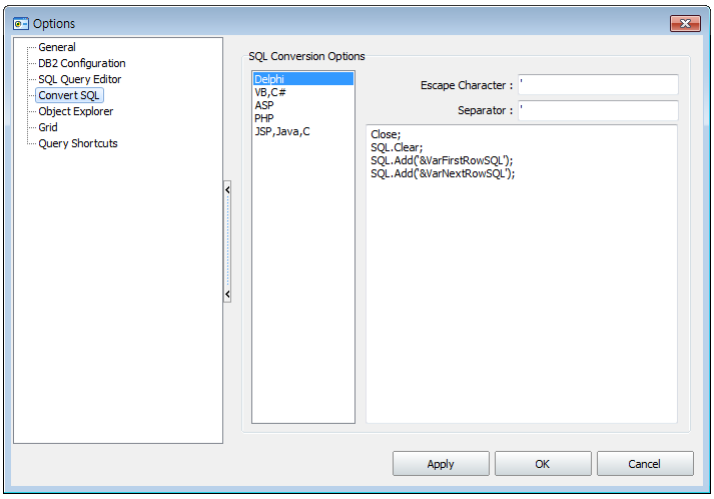

SQL Query Editor - administrator@SAMPLE30 [SQL1]
Run
Stop
SQL1*
EMPLOYEE
Result
EMPNO FIRSTNME MIDINIT LASTNAME WORKDEPT PHONENO HIREDATE JOB EDLEVEL
1 000010 CHRISTINE I HAAS A00 3978 1995-01-01 PRES
2 000020 MICHAEL L THOMPSON B01 3476 2003-10-10 MANAGER
3 000030 SALLY A JEFFERSON C01 4738 2005-04-05 MANAGER
4 000050 JOHN B ADAMSON E01 6789 1979-08-17 MANAGER
5 000060 IRVING F ADAMSON D11 6423 2003-09-14 MANAGER
6 000070 EVA D SPENSER D21 7831 2005-09-30 MANAGER
7 000090 EILEEN W SPENSER E11 5498 2000-08-15 MANAGER
8 000100 THEODORE Q SPENSER E21 0972 2000-06-19 MANAGER
9 000110 VINCENZO G LUCCHESSI A00 3490 1988-05-16 SALESREP
10 000120 SEAN O'CONNELL A00 2167 1993-12-05 CLERK
11 000130 DELORES M QUINTANA C01 4578 2001-07-28 ANALYST
12 000140 HEATHER A NICHOLLS C01 1793 2006-12-15 ANALYST
Data
Execution plan
Message History
Select
1 / 28672
1 Sec
1 : 1
Insert
Row
ANSI-DOS
Vertical Grid
28672 rc

Above example is result screen of execution of **< Alt+1 >** command after selecting 'EMPLOYEES' text.

※ SQL = Desc EMPLOYEES

- **[SQL Conversion]**

Specify using format when convert SQL. Specify Escape character and separator conversion format for each program languages.



3.7. Window

3.7.1. Tile Horizontally

Tiles windows horizontally.

Select **[Window] > [Tile Horizontally]** on the main menu.

3.7.2. Tile Vertically

Tiles windows vertically.

Select **[Window] > [Tile Vertically]** on the main menu.

3.7.3. Cascade

Tiles windows in cascade.

Select **[Window] > [Cascade]** on the main menu.

3.7.4. Arrange Icons

Arranges window icons.

Select **[Window] > [Arrange Icons]** on the main menu.

3.8. Help

3.8.1. Help

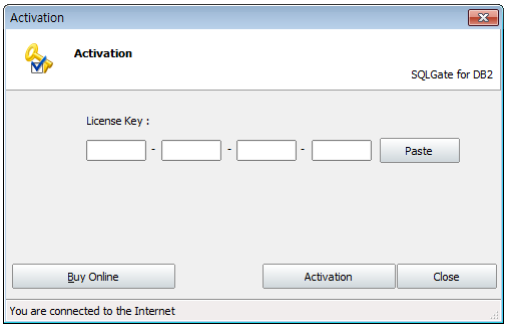
Opens help file.

Select **[Help] > [Help < F1 >]** on the main menu.

3.8.2. Activation

Authenticates program as formal commodity.

Select **[Help] > [Activation…]** on the main menu.

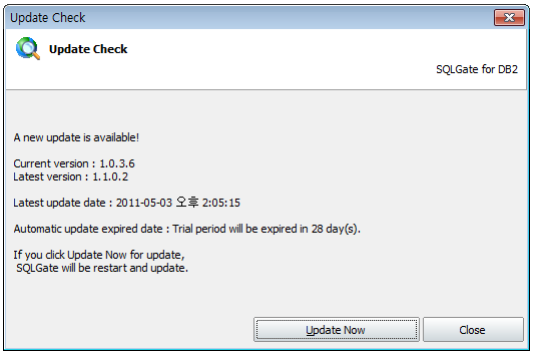


If you input the license key of 16 digits and click the **[Activation]** button, program authentication will be finished.

3.8.3. Update Check

Checks latest update information. Currently using version and latest version are displayed.

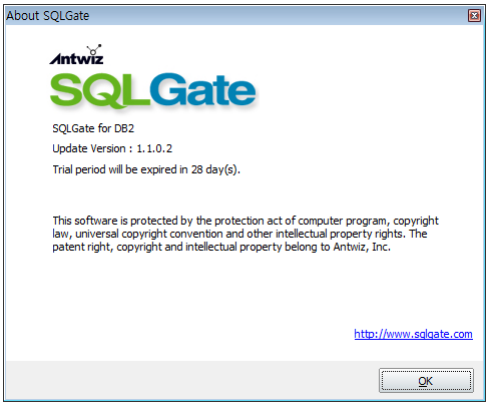
Select **[Help] > [Update Check…]** on the main menu.


Update Check
Update Check
SQLGate for DB2
A new update is available!
Current version : 1.0.3.6
Latest version : 1.1.0.2
Latest update date : 2011-05-03 오후 2:05:15
Automatic update expired date : Trial period will be expired in 28 day(s).
If you click Update Now for update,
SQLGate will be restart and update.
Update Now
Close

3.8.4. About SQLGate

Inquires program information.

Select **[Help] > [About SQLGate]** on the main menu.

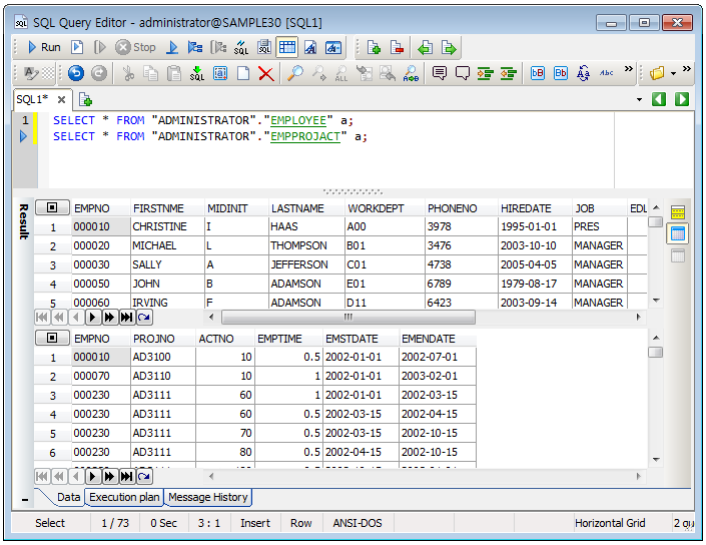

About SQLGate
Antwiz
SQLGate
SQLGate for DB2
Update Version : 1.1.0.2
Trial period will be expired in 28 day(s).
This software is protected by the protection act of computer program, copyright law, universal copyright convention and other intellectual property rights. The patent right, copyright and intellectual property belong to Antwiz, Inc.
http://www.sqlgate.com
OK

4. SQL QUERY EDITOR

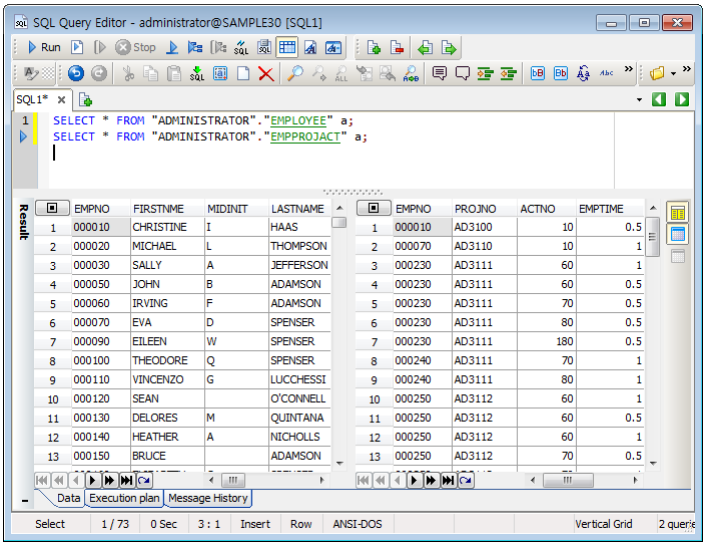
4.1. Multi-Query

You can execute several SQLs simultaneously and check results in grid.

Also, you can compare data more easily. Separator of command is (;).


SQL Query Editor - administrator@SAMPLE30 [SQL1]
Run
Stop
SQL1*
1 SELECT * FROM "ADMINISTRATOR"."EMPLOYEE" a;
2 SELECT * FROM "ADMINISTRATOR"."EMPPROJACT" a;
Result
EMPNO FIRSTNME MIDINIT LASTNAME WORKDEPT PHONENO HIREDATE JOB EDL
1 000010 CHRISTINE I HAAS A00 3978 1995-01-01 PRES
2 000020 MICHAEL L THOMPSON B01 3476 2003-10-10 MANAGER
3 000030 SALLY A JEFFERSON C01 4738 2005-04-05 MANAGER
4 000050 JOHN B ADAMSON E01 6789 1979-08-17 MANAGER
5 000060 IRVING F ADAMSON D11 6423 2003-09-14 MANAGER
EMPNO PROJNO ACTNO EMPTIME EMSTDATE EMENDATE
1 000010 AD3100 10 0.5 2002-01-01 2002-07-01
2 000070 AD3110 10 1 2002-01-01 2003-02-01
3 000230 AD3111 60 1 2002-01-01 2002-03-15
4 000230 AD3111 60 0.5 2002-03-15 2002-04-15
5 000230 AD3111 70 0.5 2002-03-15 2002-10-15
6 000230 AD3111 80 0.5 2002-04-15 2002-10-15
Data
Execution plan
Message History
Select
1 / 73
0 Sec
3 : 1
Insert
Row
ANSI-DOS
Horizontal Grid

If you click the [ ] button, you can see in horizontal grid.



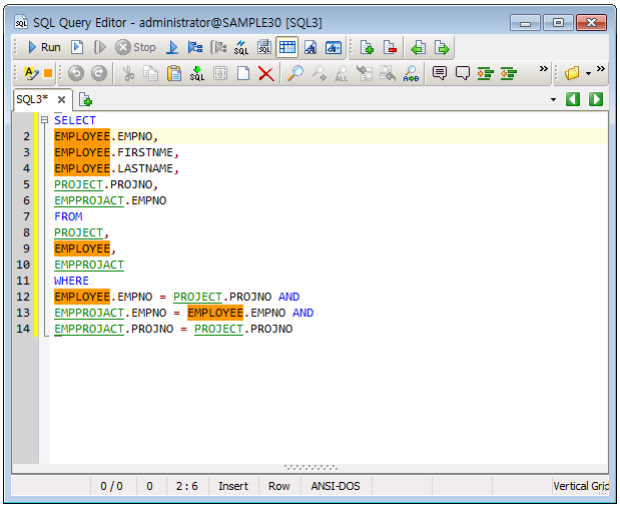
4.2. Specify background color of text

This feature marks background color of the table, which is willing to find, in SQL Query Editor.

If you select **[Find All]** when you find wanted words in the editor, you can find all same words and specify background color to found words.

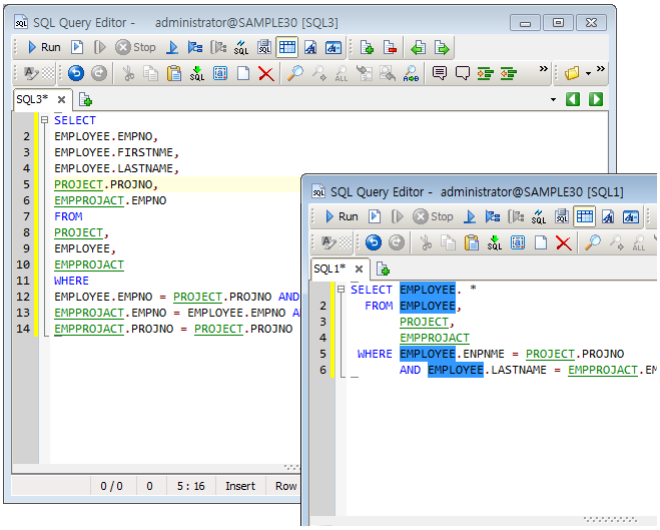
In addition, you can specify different background color to each word.

1. If you click the [ ] button after selecting the table, which is willing to see, that will be changed into wanted background color.

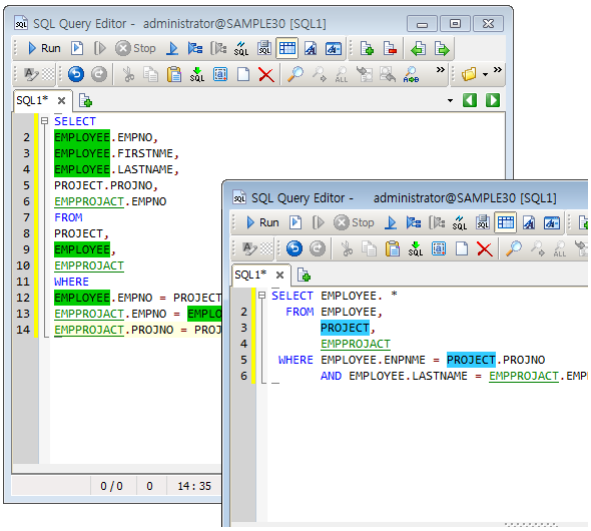


2. If you specify background color of the same object in another editor, the specified background color of the previous editor will be removed.

   According to the below image, you can see that the background color of the EMPLOYEES table in SQL1 edit or is removed when the background color of the EMPLOYEES table, which is the same object, is specified in SQL2 editor even though the background color of the EMPLOYEES table was specified in SQL1 editor.

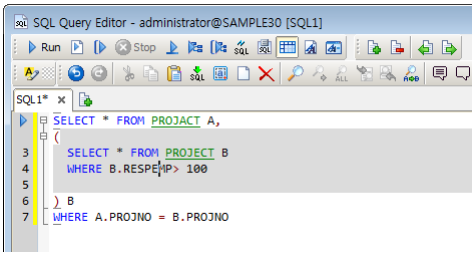


3. When you specify the background color of an object, which is not specified background color in the previous editor, in other editor, the background color will not removed in the previous editor.


SQL Query Editor - administrator@SAMPLE30 [SQL1]
Run
Stop
SQL1*
SELECT
EMPLOYEE.EMPNO,
EMPLOYEE.FIRSTNME,
EMPLOYEE.LASTNAME,
PROJECT.PROJNO,
EMPPROJACT.EMPNO
FROM
PROJECT,
EMPLOYEE,
EMPPROJACT
WHERE
EMPLOYEE.EMPNO = PROJECT
EMPPROJACT.EMPNO = EMPLO
EMPPROJACT.PROJNO = PROJ
0 / 0
0
14 : 35
SQL Query Editor - administrator@SAMPLE30 [SQL1]
Run
Stop
SQL1*
SELECT EMPLOYEE. *
FROM EMPLOYEE,
PROJECT,
EMPPROJACT
WHERE EMPLOYEE.ENPNME = PROJECT.PROJNO
AND EMPLOYEE.LASTNAME = EMPPROJACT.EMPI

4.3. Run SubQuery block

You can find subquery in complex SQL automatically and execute it.



Run subquery in the line, where cursor is located, in SQL Query Editor.

Subquery can be executed even it is not in block.

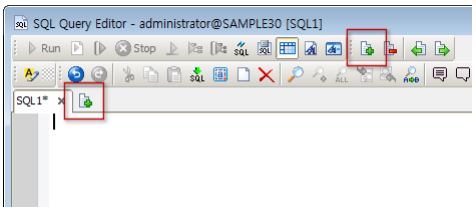
Font color of subquery in editor is changed automatically and the **[Run SubQuery block]** button will be activated.

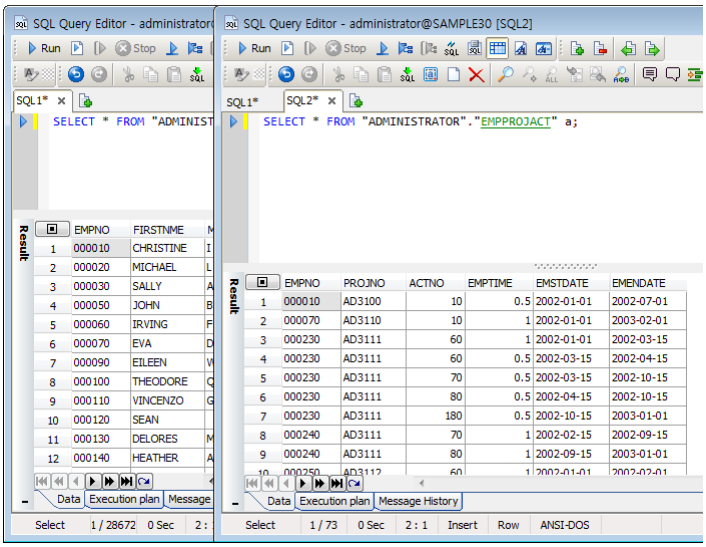
Select **[Query] > [Run SubQuery block < Alt+F5 >]** on the main menu or press the **[** **]** button.

4.4. Multi-Tab

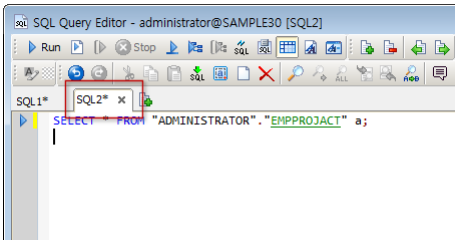
Several tabs can be added in SQL Query Editor so you don't need t open several SQL Query Editor.

Press shortcut key, < **Ctrl+Alt+N >,** or click the **[** **]** button.

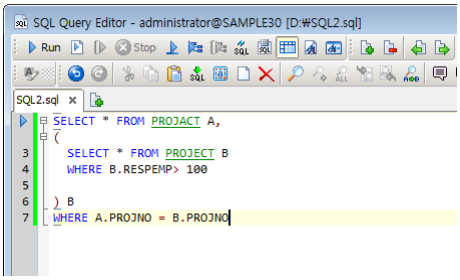




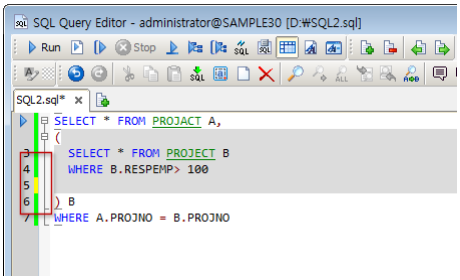
If you add or modify SQL in SQL Query Editor, (*) symbol will be displayed on the tab.



If you modify SQL after saving SQL, you can check modified line in gutter.



If you add code, yellow line is marked in gutter as below. Also, (*) symbol is displayed on the tab.

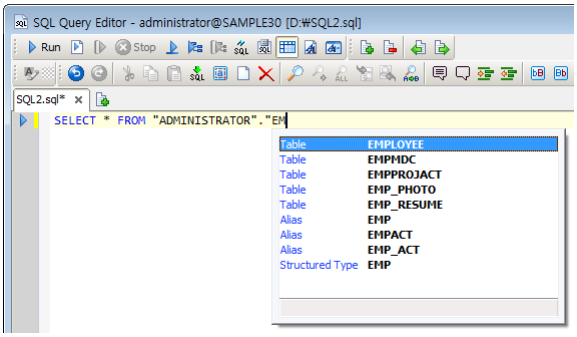


4.5. Table Auto Completion

Table lists are displayed after From/Into/Update keyword automatically.

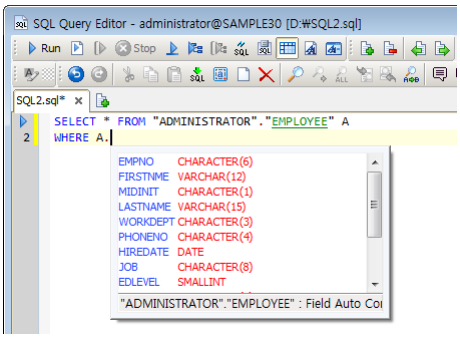
You can create SQL statement faster through this feature.

If you press shortcut key **< Ctrl+H >**, table auto completion will be executed wherever.



4.6. Field Auto Completion

If you input dot(.) in SQL Query Editor or PL/SQL Query Editor, alias or field of table will be displayed.

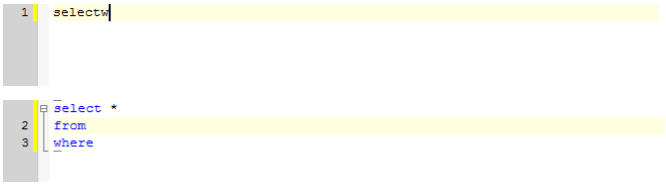


4.7. Code Auto Completion

Code Auto Completion is a useful feature that you save often using code in advance and call them whenever you need.

If you press shortcut key, **< Ctrl+J >,** code auto completion will be executed.

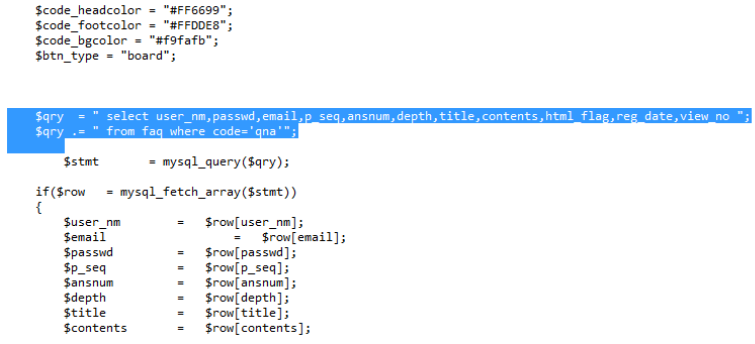
If you input **Selectw** in SQL Query Editor and then press **< Ctrl+J >** keys, it will be converted into **select * from wh ere** code.



4.8. Paste from the code

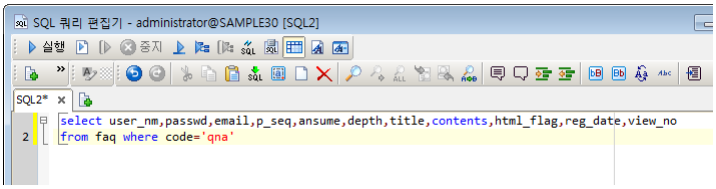
This is a feature to extract only SQL in development code.

For example, it can extract SQL statement only in PHP code that includes SQL.



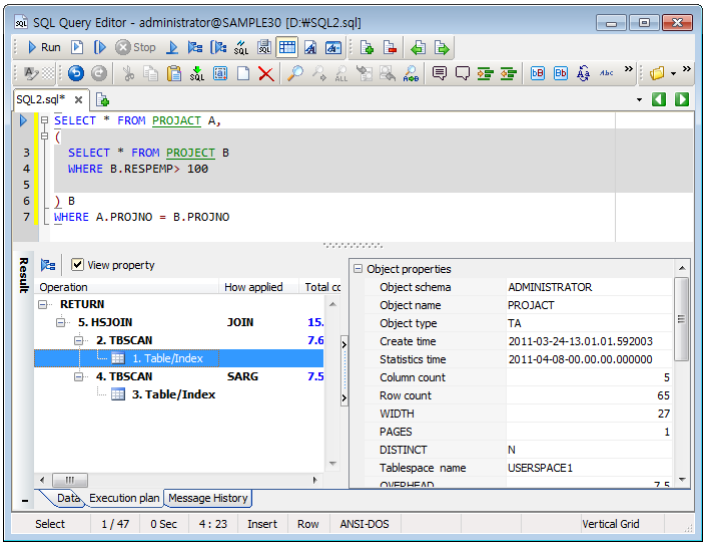
```
$code_headcolor = "#FF6699";
$code_footcolor = "#FFDDE8";
$code_bgcolor = "#f9fafb";
$btn_type = "board";


$qry  = " select user_nm,passwd,email,p_seq,ansnum,depth,title,contents,html_flag,reg_date,view_no ";
$qry .= " from faq where code='qna'";
    $stmt       = mysql_query($qry);

if($row   = mysql_fetch_array($stmt))
{
    $user_nm        =   $row[user_nm];
    $email              =   $row[email];
    $passwd         =   $row[passwd];
    $p_seq          =   $row[p_seq];
    $ansnum         =   $row[ansnum];
    $depth          =   $row[depth];
    $title          =   $row[title];
    $contents       =   $row[contents];
```

Copy above PHP code to clipboard and then select the **[Paste from the code]** on the context menu in SQL Query Editor or press shortcut key, **< Ctrl+Shift+V >**. Only SQL statement will be extracted like below.



4.9. Execution plan

If you press the shortcut key, < **F7** > or the button after creating query, execution plan results will be displayed at bottom.



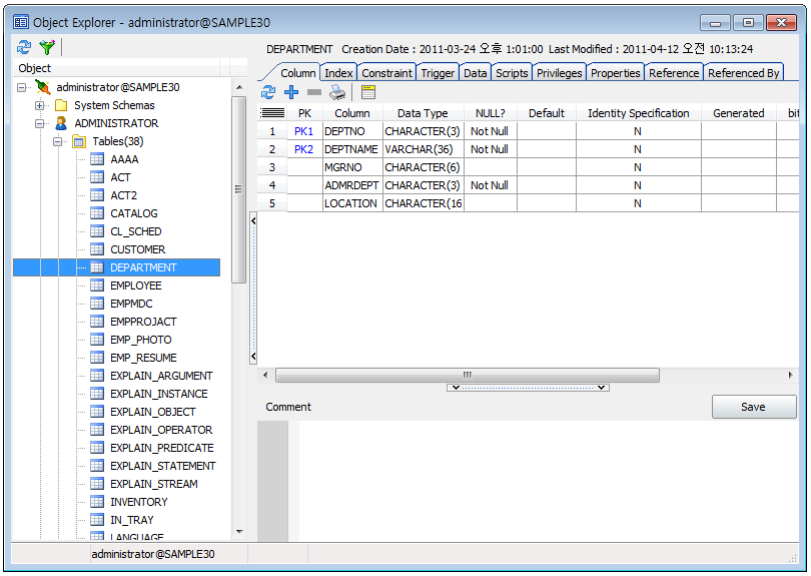
If you select a row of execution plan results, object and index information are displayed at right of the screen.

5. Object Explorer

5.1. Table

It provides overall features about tables that inquires, creates, modifies and deletes table lists by owner and detail information of each table.

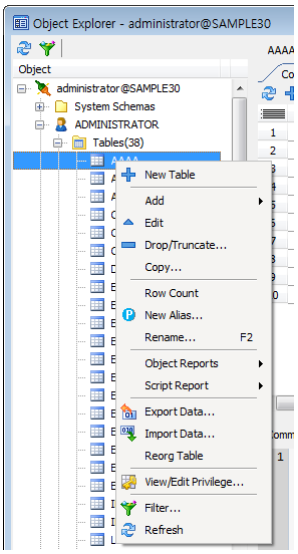
Select **[View] > [Object Explorer]** on the main menu or click the **[ ]** button.



- **Tab Information of [Table]**

- **[Column] :** You can inquire, create, delete and print out column information of the selected table. Also, you can inquire and modify column comment and table comment.
- **[Index] :** You can inquire, add, modify, delete, print out and rebuild index information of the selected table.
- **[Constraint] :** You can inquire, add, modify, delete, print out, enable and disable constraint of the selected table.
- **[Trigger] :** You can inquire, add, modify, delete, print out, enable and disable trigger information of the sel ected table.
- **[Data] :** You can inquire, add, modify, delete and print out all data of the selected table.
- **[Scripts] :** You can inquire DDL by wanting option and save it.
- **[Privilege] :** You can inquire, add, modify, delete and print out granted privilege to the selected table.
- **[Partitions] :** You can inquire partition information, if the selected table is partition table.
- **[Properties] :** You can inquire creation option of the selected table and storage information.

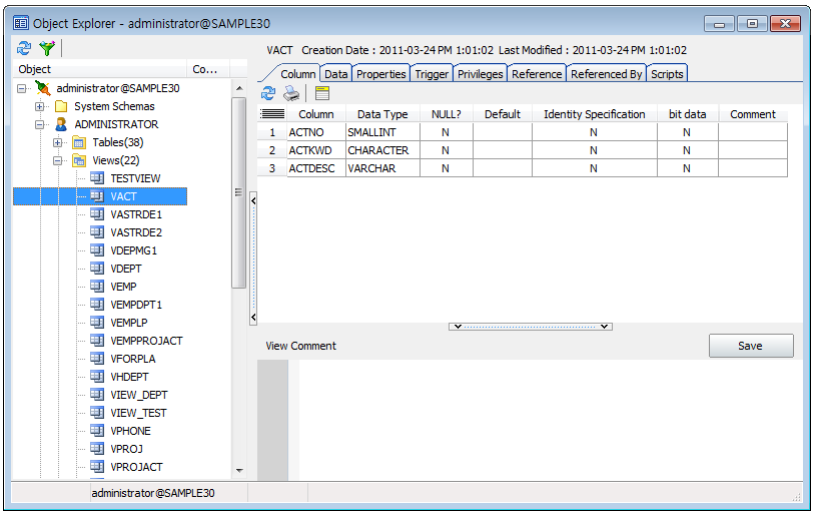
- **Context Menu of [Table]**



- **[New Table...]** : Create Table window will be popped up.
- **[Add]** :
  - **[Column...]** : Add columns to the selected table.
  - **[Constraint...]** : Add constraints to the selected table.
  - **[Index...]** : Add indexes to the selected table.
  - **[Trigger...]** : Add triggers to the selected table.
- **[Alter] :** Change column and comment of the selected table.
- **[Delete/Truncate] :** Execute Drop, Truncate command to the selected table.
- **[Copy...] :** Copy the selected table. Provides **Only Table Structure**, **Table Structure and Data**.
- **[Row Count] :** Display record count of the selected table's data.
- **[New Alias]** : Create an alias of the selected table.
- **[Rename] :** Change name of the selected table.
- **[Object Reports] :** Print out information of the selected table as reports.
- **[Script Report] :** Print out script of the selected table as reports.
- **[Export Data...] :** Export the selected table. (Excel, Csv, SQL and so on)
- **[Import Data...] :** Import the selected table. (Excel, Csv, SQL and so on)
- **[View/Edit Privilege] :** Inquire and edit privilege of the selected table.
- **[Filter…] :** Filter and sort lists by using the **Filter** window.
- **[Refresh] :** Refresh.

5.2. View

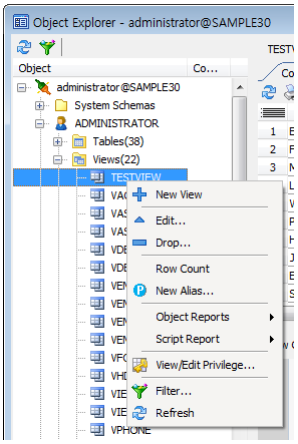
Inquires view list by owner and detail information of view and performs overall features about view, such as creation, modification, deletion, rebuilding and so on.



- **Tab Information of [View]**

- **[Column]** : You can inquire and print out column information and comment of the selected view.
- **[Data]** : You can add, modify, delete and print out data of the selected view.
- **[Properties]** : You can check properties of the selected view.
- **[Trigger] :** You can inquire, add, delete and print out trigger of the selected view.
- **[Privilege]** : You can inquire, add, delete and print out granted privilege to the selected view.
- **[Reference]** : You can inquire objects, that the view uses.
- **[Referenced By]** : You can add referenced by objects of the selected view.
- **[Scripts]** : You can inquire creation script by wanting options and save it.

- **Context Menu of [View]**



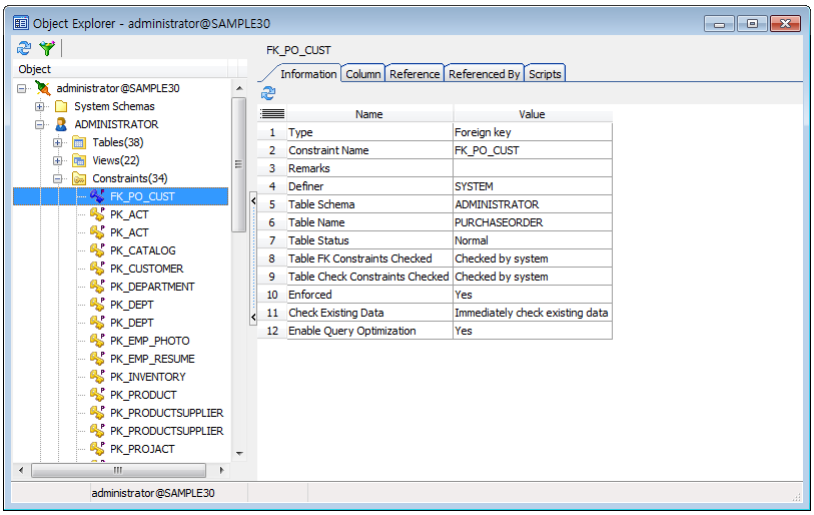
- **[New View]** : The **New View** window pops up.
- **[Alter]** : The **Alter View** window of the selected view pops up.
- **[Delete]** : Drop the selected view.
- **[Row Count]** : Show record count of the selected view's data.

-

- **[New Alias...]** : Create an alias of the selected view.
- **[Object Reports/Script Report]** : Show reports of the selected view.
- **[View/Edit Privilege]** : Inquire and edit granted privileges to the selected view.
- **[Filter…]** : Filter and sort lists by using the **Filter** window.

5.3. Constraint

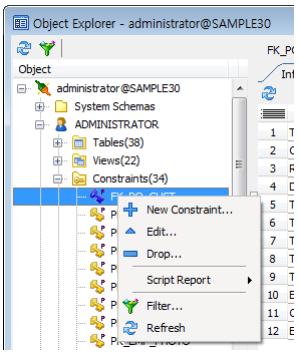
Inquires constraint list by owner and detail information of each constraint and performs overall features about constraint, such as creation, modification, deletion and so on.



- **Tab Information of [Constraint]**

- **[Information] :** You can inquire constraint information, such as owner, type, table name and so on.
- **[Column] :** You can inquire column information of constraint and print out it.
- **[Reference] :** You can check objects that the selected constraint refers to.
- **[Referenced By]** : You can add referenced by objects of the selected constraint.
- **[Scripts] :** You can inquire creation script, save to file, export to SQL Query Editor and so on.

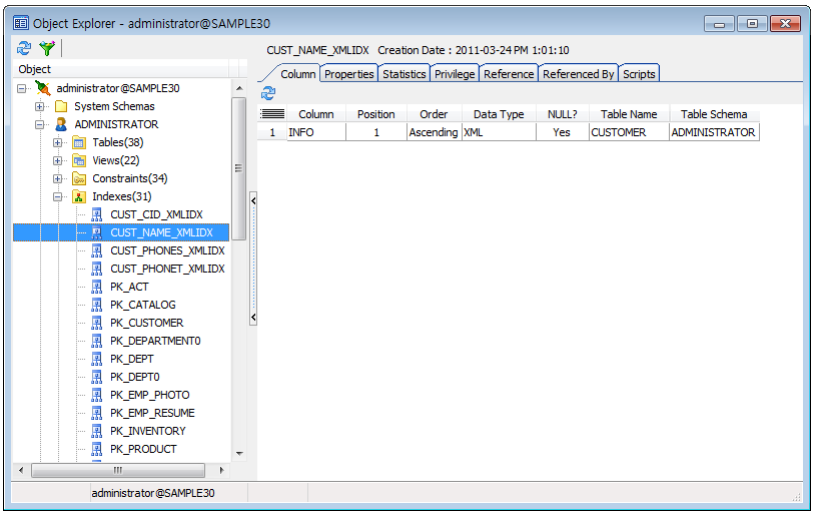
- **Context Menu of [Constraint]**



- **[New Constraint...]** : The **New Constraint** window pops up.
- **[Edit...] :** The **Edit Constraint** window of the selected constraint pops up.
- **[Drop] :** Drop the selected constraint.
- **[Script Report]** : Show reports of the selected constraint.
- **[Filter...] :** Filter and sort lists by using the **Filter** window.

5.4. Index

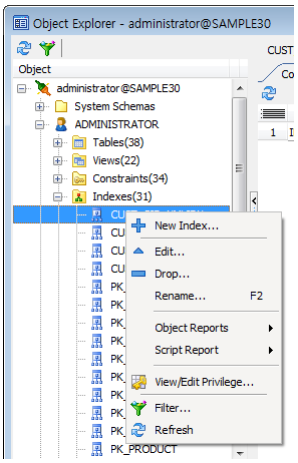
Inquires index list by owner and detail information of index and performs overall features about index, such as creation, modification, deletion and so on.



- **Tab Information of [Index]**

- **[Column] :** You can check column information of the selected index.
- **[Properties] :** You can check creation property and using storage information.
- **[Statistics] :** You can see statistics of the selected index.
- **[Privilege] :** You can inquire and edit assigned privileges to the selected index.
- **[Reference] :** You can check objects that the selected index refers to.
- **[Referenced By]** : You can add referenced by objects of the selected index.
- **[Scripts] :** You can check creation information of the selected index.

- **Context Menu of [Index]**

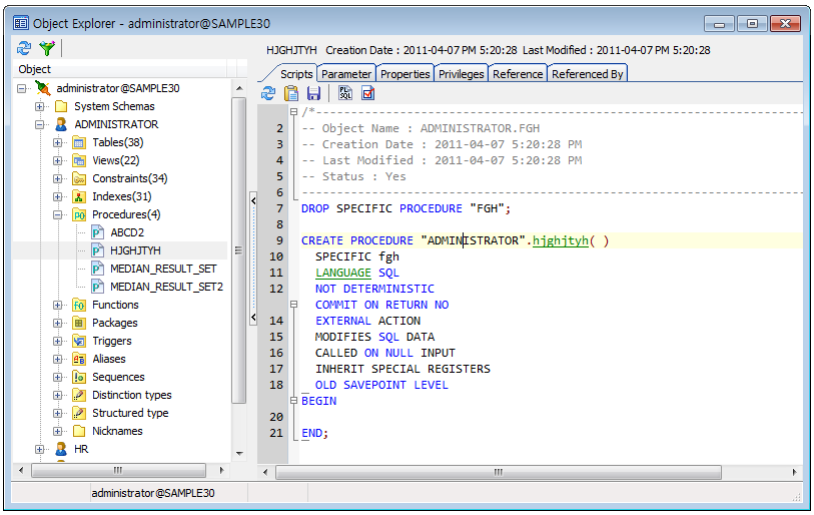


- **[New Index...]** : The **New Index** window pops up.
- **[Edit...] :** The **Edit Index** window of the selected index pops up.
- **[Delete] :** Drop the selected index.
- **[Rename] :** Change name of the selected index.
- **[Object Reports/Script Report]** : Show report of the selected index.

- **[View/Edit Privilege]** : Inquire and edit granted privileges to the selected index.
- **[Filter…] :** Filter and sort lists by using the **Filter** window.

5.5. Procedures

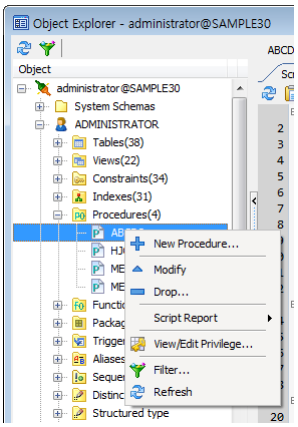
Inquires procedure list of the selected owner and detail information and performs overall features about procedure, such as creation, modification, deletion and so on.



- **Tab Information of [Procedures]**

- **[Scripts]** : Display creation script of the selected object.
- **[Parameter]** : Display parameter information of the selected object.
- **[Properties] :** Display properties of the selected object.
- **[Privilege] :** Display granted privileges to the selected procedure.
- **[Reference] :** You can check objects that the selected procedure refers to.
- **[Referenced By]** : You can add referenced by objects of the selected procedure.

- **Context Menu of [Procedures]**

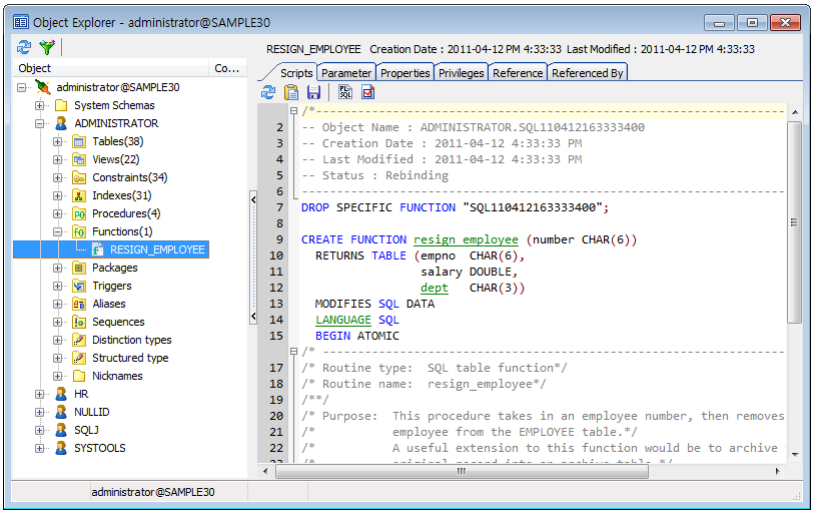


- **[New Procedures…]** : The **New Procedure** window pops up.
- **[Modify]** : Modify the selected procedure script.
- **[Drop…]** : Drop the selected object.
- **[Script Report]** : Show script reports of the selected procedure.

- **[View/Edit Privilege…]** : Inquire and edit granted privileges to the selected object.
- **[Filter…]** : Filter and sort lists by using the **Filter** window.

5.6. Function

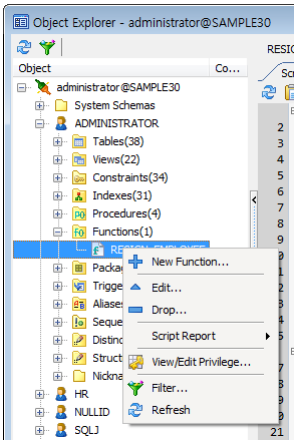
Inquires function list of the selected owner and detail information and performs overall features about procedure, such as creation, modification, deletion and so on.



- **Tab Information of [Function]**

- **[Scripts]** : Display creation script of the selected object.
- **[Parameter]** : Display parameter information of the selected object.
- **[Properties]** : Display properties of the selected object.
- **[Privilege]** : Display granted privileges to the selected function.
- **[Reference]** : You can check objects that the selected function refers to.
- **[Referenced By]** : You can add referenced by objects of the selected function.

- **Context Menu of [Function]**

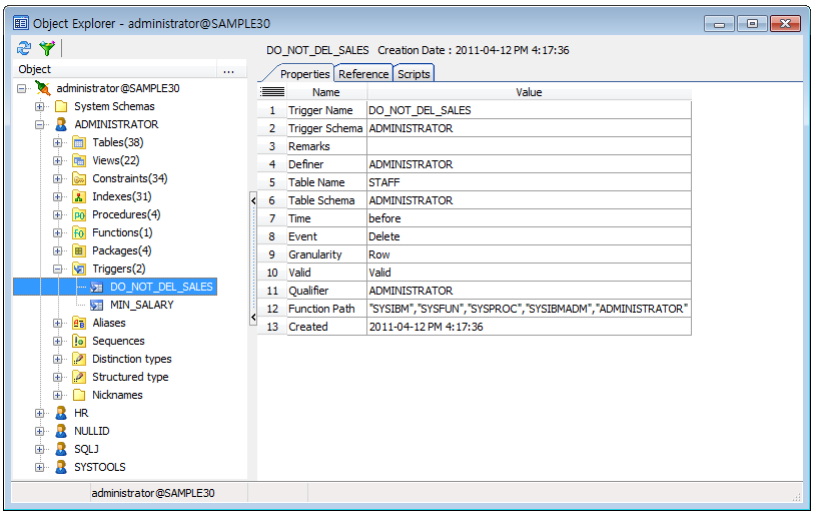


- **[New Function…]** : The **New Function** window pops up.
- **[Edit…]** : Modify the selected function script.

- **[Drop…]** : Drop the selected object.
- **[Script Report]** : Show script reports of the selected function.
- **[View/Edit Privilege…]** : Inquire and edit granted privileges to the selected object.
- **[Filter…]** : Filter and sort lists by using the **Filter** window.

5.7. Trigger

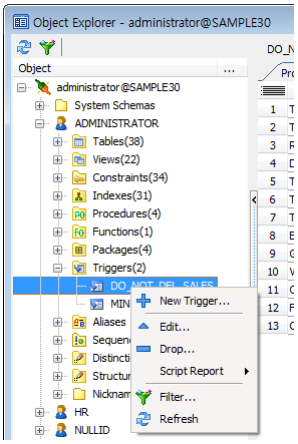
Inquires trigger list of the selected owner and detail information and performs overall features about trigger, such as creation, modification, deletion and so on.



- **Tab Information of [Trigger]**

- **[Properties]** : You can check properties of the selected trigger.
- **[Reference] :** You can check objects that the selected trigger refers to.
- **[Scripts] :** You can add referenced by objects of the selected trigger.

- **Context Menu of [Trigger]**

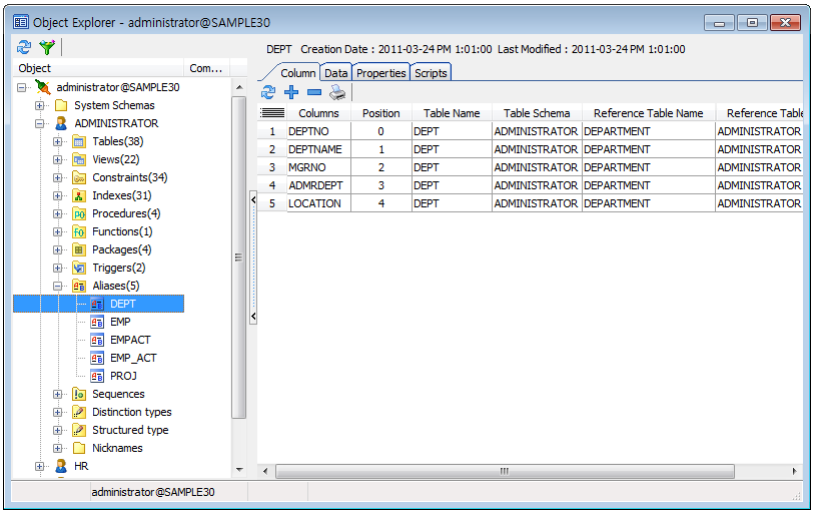


- **[New Trigger…]** : The **New Trigger** window pops up.
- **[Edit…]** : The **Edit Trigger** window of the selected trigger pops up.
- **[Drop] :** Drop the selected trigger.

- **[Script Report]** : Show script reports of the selected trigger.
- **[Filter…] :** Filter and sort lists by using the **Filter** window.

5.8. Alias

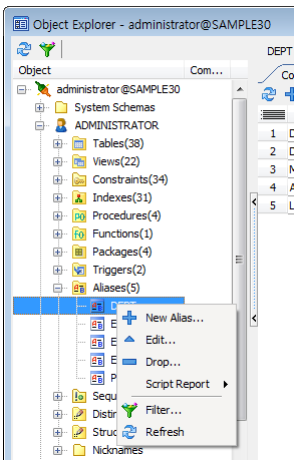
Inquires alias list of the selected owner and detail information and performs overall features about alias, such as creation, modification, deletion and so on.



- **Tab Information of [Alias]**

- **[Column]** : You can check the column of the selected alias.
- **[Data] :** You can add and modify data of the selected alias.
- **[Properties] :** You can check properties of the selected alias.
- **[Scripts] :** You can check creation script of the selected alias.

- **Context Menu of [Alias]**

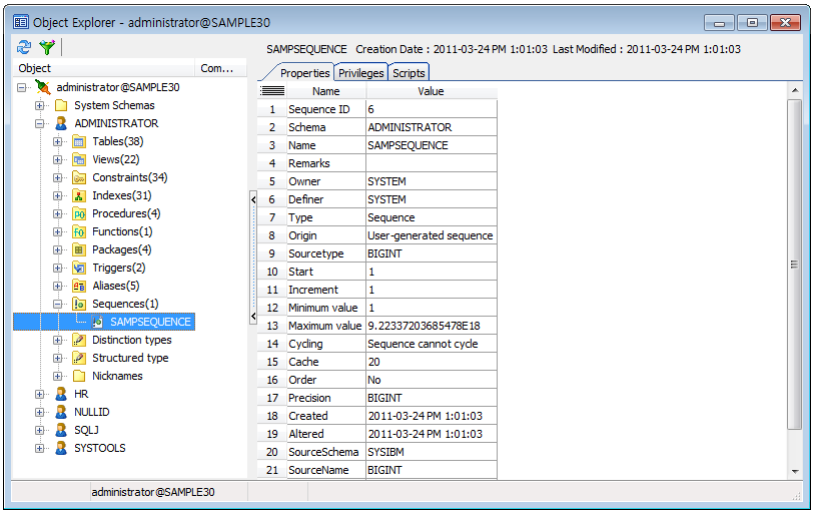


- **[New Alias…]** : The **New Alias** window pops up.
- **[Edit…]** : The **Edit Alias** window of the selected alias pops up.
- **[Drop…] :** Drop the selected alias.
- **[Script Report]** : Show script reports of the selected alias.

- **[Filter…] :** Filter and sort lists by using the **Filter** window.

5.9. Sequences

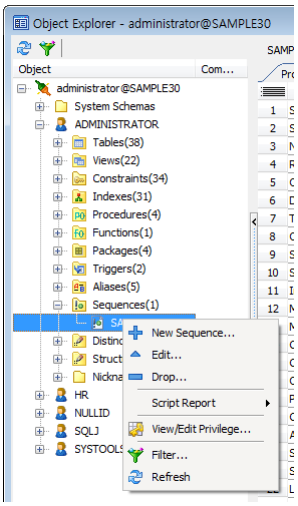
Inquires sequence list of the selected owner and detail information and performs overall features about sequence, such as creation, modification, deletion and so on.



- **Tab Information of [Sequences]**

- **[Properties] :** You can check sequence information such as, Increment by, Last, Min, Max Value and so on.
- **[Privilege] :** You can check granted privileges to the sequence.
- **[Scripts] :** You can check creation script of the selected sequence by options.

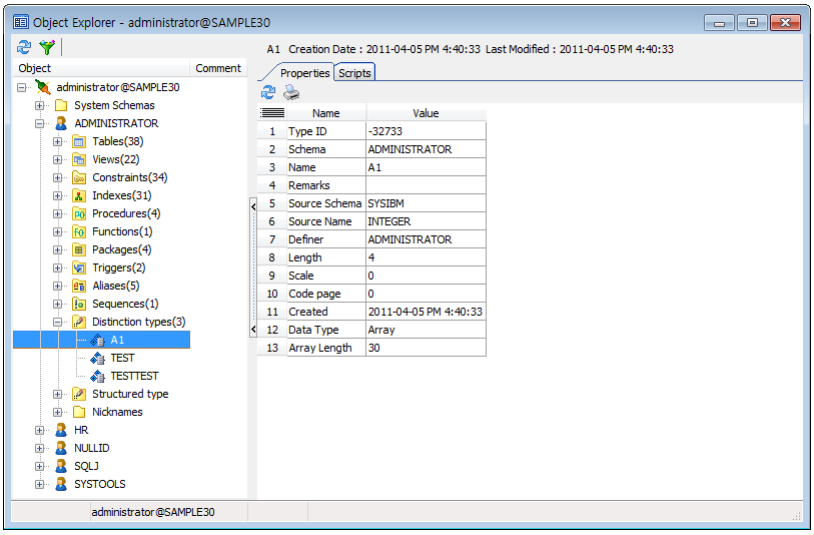
- **Context Menu of [Sequences]**



- **[New Sequences…]** : The **New Sequence** window pops up.
- **[Edit…]** : The **Edit Sequence** window of the selected trigger pops up.
- **[Drop…]** : Drop the selected sequence.
- **[Script Report]** : Show script reports of the selected sequence.
- **[View/Edit Privilege]** : Inquire and edit granted privileges of the selected sequence.
- **[Filter…]** : Filter and sort lists by using the [**Filter**] window.

5.10. Distinct type

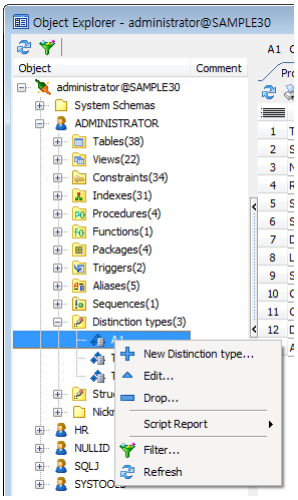
Inquires object type and collection type list by the selected owner and detail information of each types and performs overall features about type, such as creation, modification, deletion and so on.



- **Tab Information of [Distinct type]**

- **[Properties] :** You can inquire properties list of the selected Distinct type.
- **[Scripts] :** You can inquire the creation script, save to file and send to SQL Query Editor.

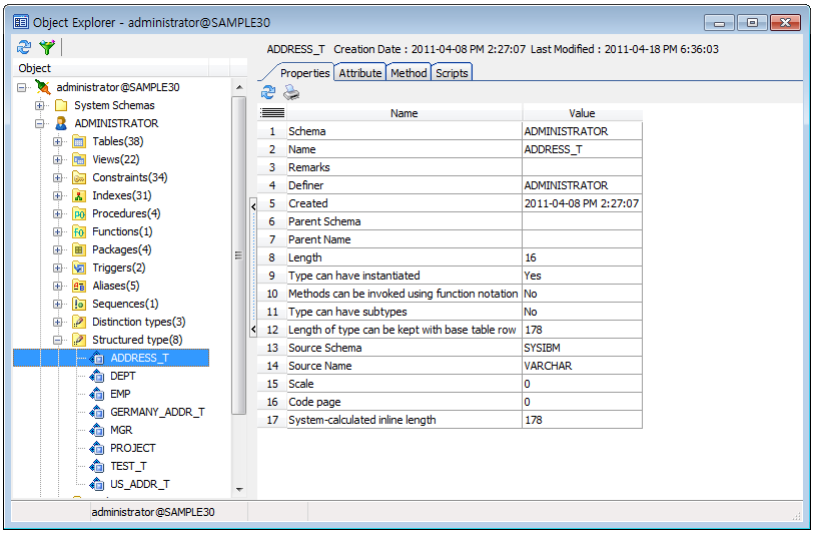
- **Context Menu of [Distinct type]**



- **[New Distinct type...] :** The **New Distinct type** window pops up.
- **[Edit...]** : The **Edit Distinct type** window of the selected Distinct type pops up.
- **[Drop...] :** Drop the selected Distinct type.
- **[Script Report]** : Show script reports of the selected Distinct type.
- **[Filter...] :** Filter and sort lists by using the **Filter** window.

5.11. Structured type

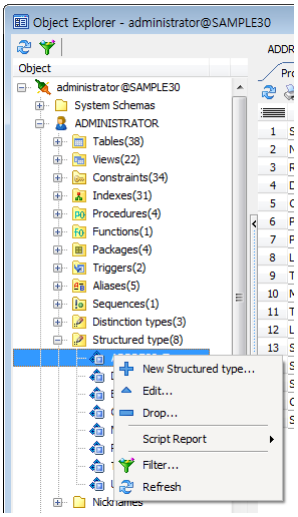
Inquires structured type list by the selected owner and detail information and performs overall features about structured type, such as creation, modification, deletion and so on.



- **Tab Information of [Structured type]**

- **[Properties]** : You can inquire properties list of the selected type.
- **[Attribute]** : You can inquire attribute list of the selected type.
- **[Method] :** You can inquire method list of the selected type.
- **[Scripts] :** You can inquire the creation script, save to file and send to SQL Query Editor.

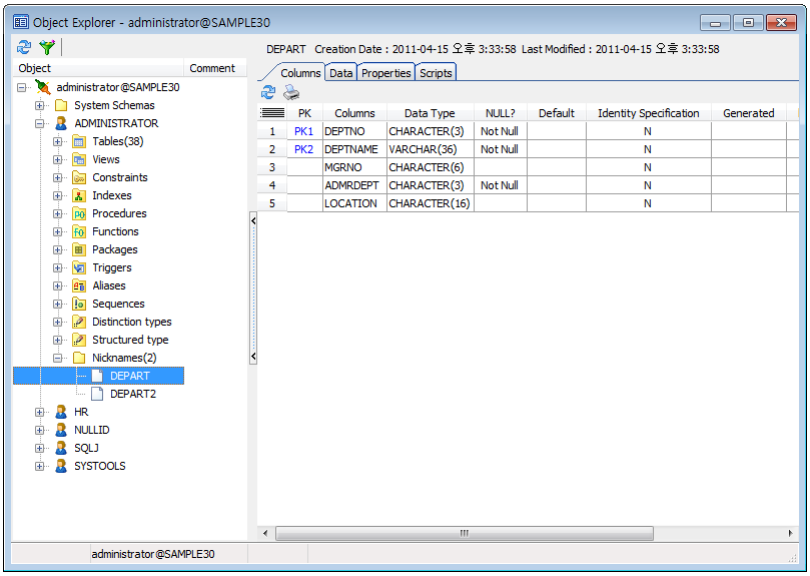
- **Context Menu of [Structured type]**



- **[New Structured type...] :** The **New Structured Type** window pops up.
- **[Edit...]** : The **Edit Structured Type** window of the selected structured type pops up.
- **[Drop...] :** Drop the selected structured type.
- **[Script Report]** : Show script reports of the selected structured type.
- **[Filter...] :** Filter and sort lists by using the **Filter** window.

5.12. Nickname

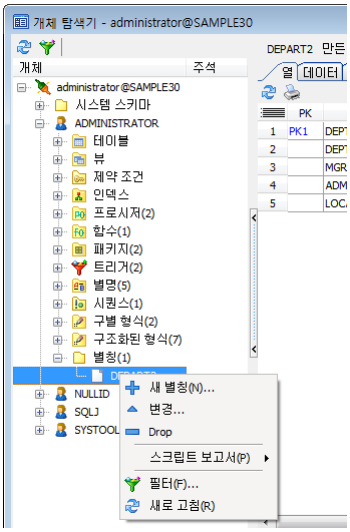
Inquires nickname list by the selected owner and detail information and performs overall features about nickname, such as creation, modification, deletion and so on.



- **Tab Information of [Nickname]**

- **[Column]** : You can inquire column information of the selected nickname.
- **[Data] :** You can add and modify data of the selected nickname.
- **[Properties]** : You can inquire properties list of the selected nickname.
- **[Scripts] :** You can inquire creation script by options.

- **Context Menu of [Nickname]**



- **[New Nickname...]** : The **New Nickname** window pops up.
- **[Edit...]** : The **Edit Nickname** window of the selected nickname pops up.
- **[Drop...] :** Drop the selected nickname.
- **[Script Report]** : Show script reports of the selected nickname.
- **[Filter...] :** Filter and sort lists by using the **Filter** window.

6. Query Builder

You can create quires only by draging&dropping easily and check results by using Query Builder.

You can join, sort and filter query objects (query, view, column and so on) with simple mouse operation and check results by creating various queries.

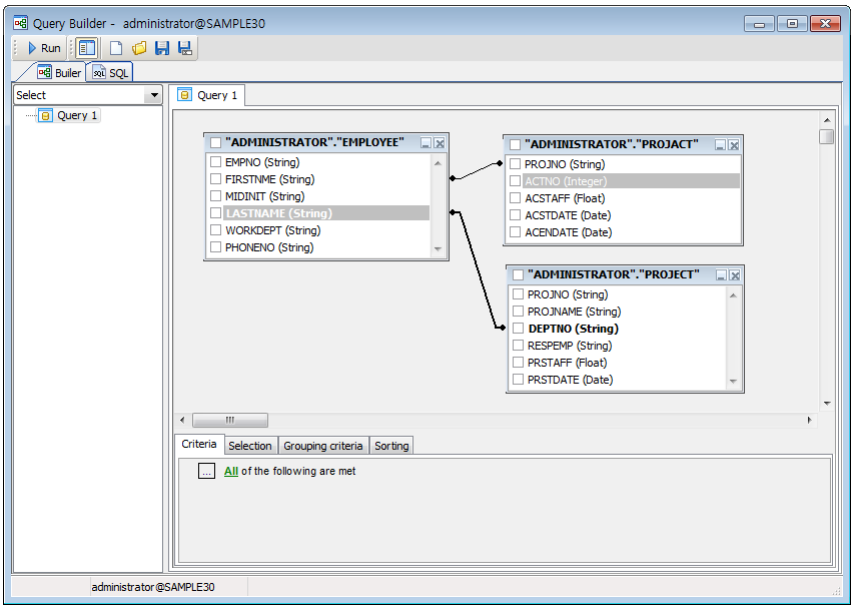
Select **[File] > [New] > [Query Builder]** on the main menu or click the [ ] button on the toolbar.

- **Select Table/View**

Open the Object Panel by selecting **[View] > [Object Panel]** on the main menu or clicking the [ ] button on the toolbar.

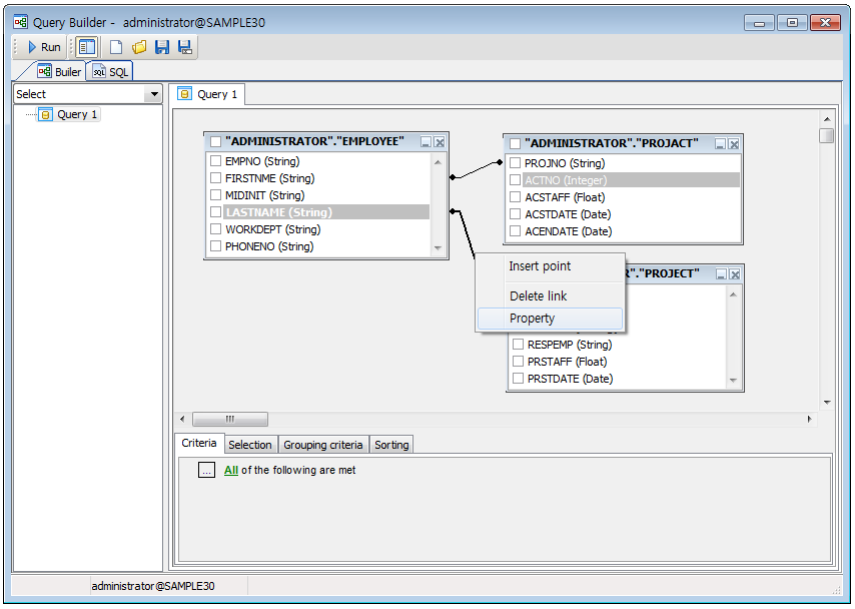
Select needed table or view in object panel and drag&drop at query builder by using mouse.

Check columns to display in SQL.

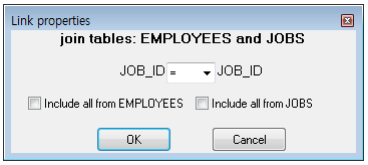


- **Join**

Select a line between objects, that join was established and then click the **[Property]** on the context menu.



Select **<Inner Join>/<Outer Join>**.



- **Set Where, Select, Group Sorting Condition**

You can add, modify and delete Where, selection, group and sorting condition at the bottom of query builder.

- **View SQL**

If you select the **[SQL]** tab, you can check the created SQL.

- **Run**

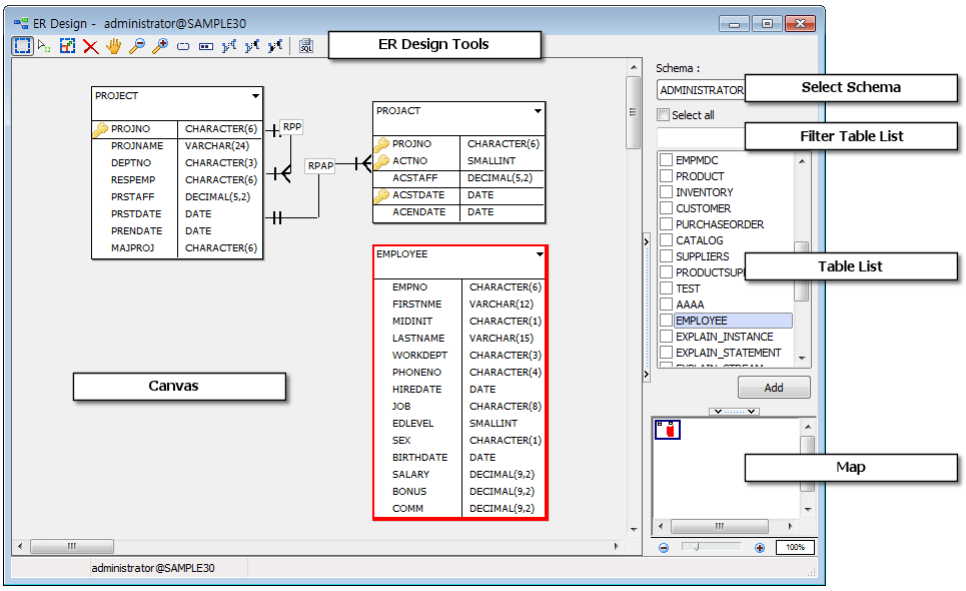
Press the shortcut key, **<F5>**, or click the [ ] button. SQL is executed and you can check results.

7. ER Design

You can create ERD (entity-relationship diagram) by applying Reverse Engineering to relationship between tables in ER Design window. It provides basic features, which are necessary for ERD edit, such as group area, text box and so on also provides ERD print, file saving and importing.

Select **[File] > [New] > [ER Design]** on the main menu or click the [ ] button.

- **Screen Composition**



- **ER Design Tools**

[Select/] : It is a feature to select object in drag area by dragging mouse.

[Move/] : Move the selected object.

[Resize/] : Resize group area.

[Delete/] : Delete the selected object.

[Panning Hand/] : Move working area by dragging mouse.

[Zoom Out/] : Zoom out **Canvas**.

[Zoom In/] : Zoom in **Canvas**.

[Region/] : Add the **[Region]** object, which ties up objects into one group.

[Text Box/] : Add sentence with simple contents.

[1:1 Non-Identifying-Relationship/ **]** : Set the **[1:1 Non-Identifying-Relationship]** between table objects.

[1:n Non-Identifying-Relationship/ **]** : Set the [**1:n Non-Identifying-Relationship]** between table objects.

[n:m Non-Identifying-Relationship/ **]** : Set the [**n:m Non-Identifying-Relationship]** between table objects.

[View SQL/ **]** : Inquire creation script of currently designed ERD.

- **Canvas**

This is working space that you can place ERD objects (Table, Text Box, Region Object) on the screen manually. You can design ERD in easier way.

- **Select Schema**

Tables of the selected database in select schema are displayed on table list.

- **Filter Table List**

This is a feature to filter table list and it shows only table name, which has inputted word, in table list.

- **Select All**

Check all checkboxes by table in table list.

- **Table List**

Display only table, which has filter word, among tables of the selected schema in select schema.
If there is no filtering word, all table lists of the selected database are displayed.

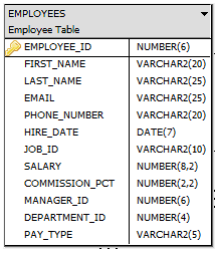
- **Add Table**

If you drag to **Canvas** or double click it after selecting a table in table list, the selected table will be revered and putted on **Canvas**.
If you click the **[Add]** button, checked tables in table list will be reversed and putted on **Canvas**.

- **Map**

- Show table on Canvas and currently working area briefly. You can move working area of Canvas by clicking the mouse or dragging.
- If you click the **[Zoom Out]** button, contract **Canvas**.
- If you click the **[Zoom In]** button, expands **Canvas**.

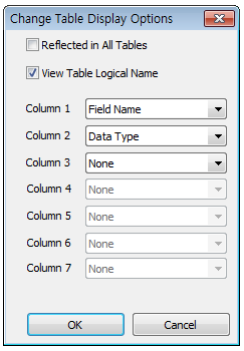
7.1. Table Object



This is an added object on **Canvas** after that you select a table from **table list.**

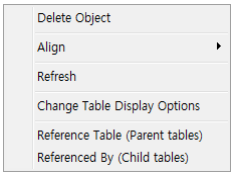
Displays column list of table, primary key, data type, logical field name and so on.

7.1.1. Table Object Menu



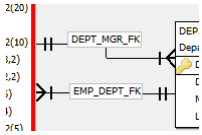
Click the right mouse button at table object and then select the [**Change Table Display Options**] on the context menu.

- **Reflected in All Tables** : Reflect changed option value of table display to all tables.
- **View Table Logical Name** : Display with logical name of table.
- **Column1 ... Column n** : Select items to display at table object.
- **[OK]** : Reflect changed option value of table display to table.



- **Delete Object** : Delete the selected object.
- **Align**: Align the table to up, down, left and right / horizontal, vertical.
- **Refresh** : Refresh the table.
- **Reference table (Parent table) / Referrer (Child table)** : If there is a constraint on the table, the parent or the child table of the selected table will be called automatically.

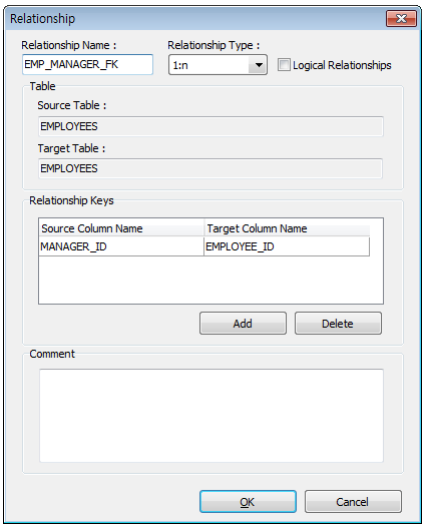
7.2. Relationship



If you select one among **1:1 Non-Identifying-Relationship /** **, 1:n Non-Identifying-Relationship /** **, n:m Non-Identifying-Relationship /** on the [ER Design] tool after that click original table object and target table object in order, logical relationship of tables will be set.

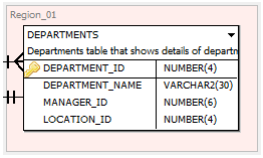
7.2.1. Change Relationship object

If you double click the relationship object, the Change Relationship window will pop up.



- **Relationship Type** : Select one among 1:1 / 1:n / n:m.
- **Logical Relationships** : When checked, it displays as dotted line.
- **Source Table** : Original table of relationship is displayed.
- **Target Table** : Target table of relationship is displayed.
- **Relationship Keys** : Relationship column lists between original table and target table are displayed.
  - **Source Column Name** : Column list of original table, which is related to column of target table, is displayed.
  - **Target Column Name** : Column list of target table, which is related to column of original table, is displayed.
- **Add** : If you click the [Add], you can input columns, which are going to be target of relationship key, additionally.
- **Delete** : If you select relationship key list and then click the [Delete], the selected relationship key will be deleted.
- **Comment** : Input description of relationship.

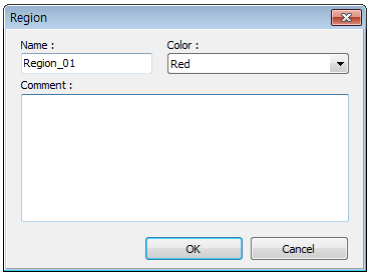
7.3. Group Area



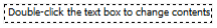
Ties up table, text objects to one group.

Object in group area moves with group area when moves group area.

Modify **Name/Color/Comment** after popping up Edit Region window by double clicking.

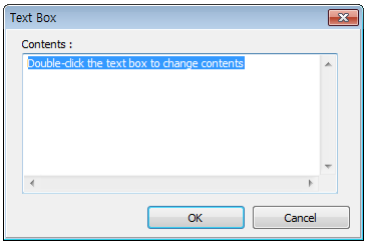


7.4. Text Box

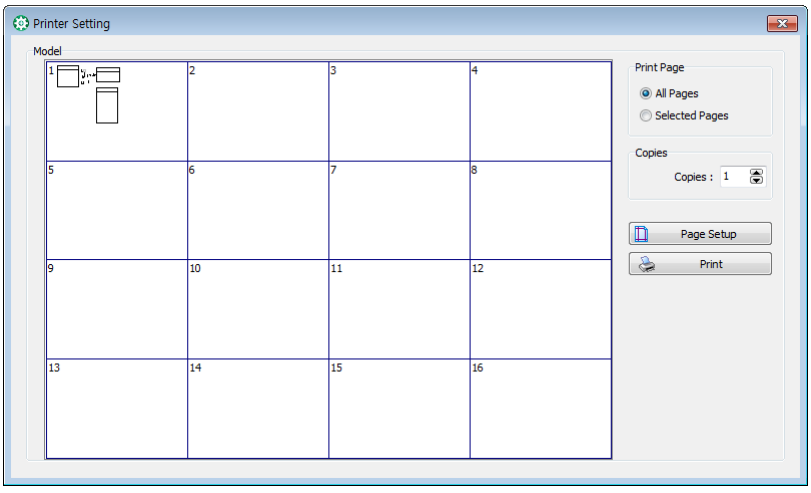


Add sentence of simple contents on **Canvas**.

Open the text box edit window by double clicking and then edit contents in text box.



7.5. Setup Print



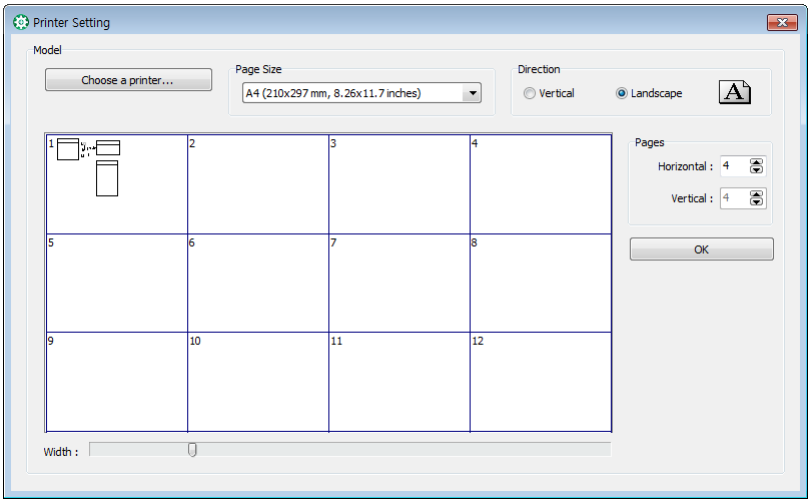
Select **[File] > [Print]** on the main menu to open the Setup Print window.

- **Print Page**

- **All Pages :** Print out whole model.
- **Selected Pages :** Print the selected model in model.

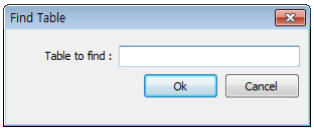
- **Model** : Select only a model, that you want to print out, with mouse in status pressing the < Ctrl> key.
- **[Page Setup]** : Page setup window shows up.
- **[Print...]** : Print model through printer.

7.5.1. Page Setup



- **[Choose a printer...]** : Select a printer to print out after popping the Choose a Printer window up.
- **Page Size** : Select paper to print.
- **Direction** : Select orientation of print paper, vertical or landscape.
- **Pages**
  - **Landscape** : Select how many pages to print widthwise.
  - **Vertical** : Select how many pages to print palewise.
- **Width** : Set width to print on print paper by dragging mouse.
- **[Printer Setting]** : Back to previous window.

7.6. Find Table Object



Select **[Edit] > [Find and Replace] > [Find]** on the main menu or press the shortcut key, **< Ctrl+F>**, to open the Find window.

Input a word to search among table names at the **Keyword**.

If you click the **[OK]**, it finds out that table by moving working area of Canvas after finding table object, which has keyword. Press the < **F3 >** key to find next table.

8. Trivia

8.1. Table of shortcut keys

Common	New	Ctrl + N	Cut	Ctrl + X
	Open	Ctrl + O	Copy	Ctrl + C
	Save	Ctrl + S	Paste	Ctrl + V
	Save As	Ctrl + Shift + S	Select all	Ctrl + A
	Undo	Ctrl + Z	Delete	Del
	Redo	Shift + Ctrl + Z		
Edit	Comment Block	Ctrl + -	Capitalize	Ctrl + I
	Uncomment Block	Shift + Ctrl + -	Increase Indent	Tab
	Upper Case	Ctrl + U	Decrease Indent	Shift + Tab
	Lower Case	Ctrl + L	Description Schema	F4
Search	Find	Ctrl + F	Find in Files	Shift + Ctrl + F
	Find Next	F3	Toggle Bookmark	Ctrl + Alt + Num
	Replace	Ctrl + R	Goto Bookmark	Ctrl + Num
	Paste from the code	Shift + Ctrl + V	Goto line	Ctrl + G
SQL HISTORY	SQL History	F8	Copy to Clipboard	Ctrl + C
	Delete SQL	Del	Open File	Ctrl + O
	Append to editor	Ctrl + Enter		
Query Editor	Run SQL	F5	Run SubQuery block	Alt + F5
	Stop	ESC	Run Current SQL and Modify	F6
	Close	Ctrl + Alt + D	Execution plan of current SQL	F7
	Add Editor	Ctrl + Alt + N	Execution plan Block SQL	Alt + F7
	Previous Editor	Ctrl + Alt + Left	Query Description	Shift + F4
	Next Editor	Ctrl + Alt + Right	Add to User SQL	Shift + F8
	Open File	Ctrl + O	SQL Align	F9
	Save	Ctrl + S	Table Auto Completion	Ctrl + H
	Print Preview	Ctrl + P	Paste from the code	Shift + Ctrl + V
	Show Grid	F11	SQL Row Count	F12
	Run Current SQL	Ctrl + F5		
Options	Code Auto Completion / Template	Ctrl + J	Reuse SQL	Ctrl + M
	Keyword Auto Completion	Ctrl + K	Field Auto Completion	.
Help	Help	F1		